National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

Canada

THE UNIVERSITY OF ALBERTA

RECURSIVE MODELS AND CONTROLLERS OF FLEXIBLE MANIPULATORS

by

JOEL OSCRETE KING

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1988

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-45453-9

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR JOEL OSCRETE KING

TITLE OF THESIS RECURSIVE MODELS AND CONTROLLERS OF FLEXIBLE MANIPULATORS

DEGREE FOR WHICH THESIS WAS PRESENTED DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED FALL, 1988

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Joel King

PERMANENT ADDRESS:

540 Michener Park
Edmonton, Alberta
Canada, T6H 4M5

DATED 13th October 1988

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled RECURSIVE MODELS AND CONTROLLERS OF FLEXIBLE MANIPULATORS submitted by JOEL OSCRETE KING in partial fulfilment of the requirements for the degree of DOCTOR OF PHILOSOPHY.

V. G. Gourishankar
.....

Supervisor

Roger W Toogood
.....

Xiaobo Li
.....

R. E. Rink
.....

M S Dare
.....

External Examiner

Date October 12 1988

ABSTRACT

Conventional robot manipulators are built with links that have relatively large diameters in order to prevent link vibrations during the manipulator motion. This practice facilitates design of dynamic control systems in that link-flexibility need not be included in the already complex dynamic equations of the manipulator. However, the heavy links that result from making link cross-sections large require larger actuators to move them. In applications where fast motion is desired, the large manipulator mass presents an obstacle to achieving the desired rate of motion. In addition, high-powered actuators tend to be quite expensive. Energy efficiency also suffers.

There is now a need for lightweight arms that can move relatively fast with low-powered actuators. The reduction in mass is obtained by reducing the diameter of each link. Flexibility must now be included in the models on which dynamic control systems are based. Link flexibility has its own advantages in that active compliance with external objects is now possible. Modeling and control of flexible manipulators is a relatively recent research discipline. Accurate methods for including flexibility in manipulator models exist but they tend to be inefficient from a computational standpoint. Furthermore, control systems for flexible manipulators have so far been proposed only for arms with one or two links, and which move in only one plane.

In this thesis, fast and efficient computational algorithms

for modeling general, flexible manipulators are presented. These algorithms are recursive in nature, and they possess attractive computational features that make it easy to design special-purpose computing structures for their implementation. Both modal and finite element approaches to link kinematics are employed. The models are quite accurate. Also, both inverse and forward dynamics are considered in the thesis.

A modified "computed torque" method of controlling multi-link, spatial, flexible manipulators is presented and tested by simulation of a three link manipulator that moves in three dimensional space. The method is general and can be applied directly to manipulators with arbitrary configurations. Results show that good end-point trajectory following can be achieved with low powered actuators. Speed of motion can be significantly increased compared with rigid manipulators that are equipped with actuators of the same power.

Finally, the computational requirements of the control algorithms are discussed and it is shown that the algorithms can be cast into linear recurrence form for easy implementation on computers with parallel processing capabilities.

ACKNOWLEDGEMENTS

The author would like to express his deep gratitude to his supervisor, Professor V.G. Gourishankar, for his guidance in formulating the objectives of this work, and for his encouragement and support throughout the research. It was his wise counsel that brought the author to this research in the first place. His enormous patience and personal support have greatly encouraged the author in this and other endeavours.

The author is heavily indebted to his co-supervisor, Professor R.E. Rink, for his guidance in the past and for his many helpful suggestions and constructive criticisms made during the preparation of this thesis. His genuine interest, patience and encouragement are most gratefully acknowledged.

The author also wishes to thank Dr. R. Toogood for his invaluable assistance as part of the author's supervisory committee. His guidance and suggestions were also instrumental in bringing the author to this specific area of research, and his continued encouragement and helpful suggestions are greatly appreciated.

The many useful discussions held with other graduate students, in particular, Mr. Frank Niscak, have been a source of guidance and encouragement. The author would like to express his thanks for these discussions and to gratefully acknowledge the benefit that they have been to him.

The financial support the author received from the Department of Electrical Engineering, in terms of teaching assistanships and stipends from supervisors' research funds, is gratefully acknowledged. This research would not have been possible without such support. Also, the use of the department's HP9000 computer system to perform the simulations is sincerely appreciated. The assistance received from the system administrator, Mr. Steve Drake, has been of untold benefit and his patience is gratefully acknowledged. The author would also like to thank the Department of Computing Science for use of their computing system during the early part of this research.

Finally, the forebearance of the author's family during the period of this research cannot go without mention. The sacrifices that were made by them are too numerous to mention. The author owes to them a debt of gratitude that would never be repaid in full.

TABLE OF CONTENTS

Chapter ..... Page

1. OVERVIEW OF THE RESEARCH PROJECT 1

1.1 Introduction 1

1.2 Organization of the Thesis 4

1.3 Background Material 5

1.4 Justification of Flexibility 5

1.5 Modeling of Robot Manipulators 8

1.5.1 Kinematics of Rigid Manipulators 8

1.5.2 Kinematics of Flexible Links 13

1.6 Manipulator Dynamics 17

1.7 Dynamics of Flexible Manipulators 21

1.8 Contributions of the Thesis to Flexible Manipulator Modeling 23

1.9 Modern Approaches to Manipulator Dynamic Control 25

1.9.1 Rigid Manipulators 25

1.9.2 Flexible Manipulators 27

2. RECURSIVE ASSUMED MODES MODELING OF FLEXIBLE MANIPULATORS 30

2.1 Introduction 30

2.2 Calculation of Link Velocities and Accelerations 31

2.3 Formulation of the System Kinetic Energy 35

2.4 Equations of Motion of the Flexible Manipulator 39

2.4.1 Lagrangian Equations of Motion 39

2.4.2 Joint Equations of Motion 40

2.4.3 Flexibility Equation 46

2.4.4 Elastic Potential Energy Term.....51
2.5 Computational Algorithm for the Inverse Dynamics.....54
2.6 Computational Algorithm for Dynamic Simulation.....60
2.7 Simulation Example.....65
2.8 Comparison with Book's Recursive Lagrangian Method.....67
2.9 Discussion and Summary.....74

3. RECURSIVE FINITE ELEMENT MODELING OF FLEXIBLE MANIPULATORS...77
3.1 Introduction.....77
3.2 Finite Element Kinematics.....79
3.3 Boundary Conditions.....86
3.4 Kinetic Energy Computations.....87
3.5 Potential Energy Computation.....92
3.6 Inverse Dynamic Equations.....93
3.7 Recursive Algorithm for Inverse Dynamics Calculation.....97
3.8 Recursive Algorithm for Calculation of Forward Dynamics.....101
3.9 Simulation Example.....107
3.10 Comparison Between Finite Element and Assumed Modes Models.....112

4. COMPOSITE CONTROL OF FLEXIBLE MANIPULATORS.....122
4.1 Introduction.....122
4.2 Computed Torque Control of Rigid Manipulators.....123
4.3 Computed Torque Control of Flexible Mnaipulators.....126
4.4 System Decomposition.....128

4.5 Example Arm.....132
4.6 Simulation Results with Joint Controller Only.....134
4.7 Design of the Pseudolink Controller.....139
4.8 Simulation Results.....148
4.9 Implementational Consideration.....157
4.9.1 Sampling Rate Considerations.....158
4.10 Conclusion.....163

5. RECURRENCE RELATIONS AND PARALLELISM IN FLEXIBLE MANIPUALTOR DYNAMICS.....166
5.1 Introduction.....166
5.2 Computational Requirements of the Composite, Pseudolink Controller.....166
5.3 Parallelism in Linear Recurrence Problems.....169
5.4 Inverse Dynamics of Flexible Manipulators as Linear Recurrence.....172
5.4.1 Model for Calculating the Non-Recursive Equations.....173
5.4.2 Parallel Algorithm for Calculating the Recursive equations.....182
5.5 Forward Dynamics of Flexible Manipulators as Linear Recurrences.....186
5.6 Conclusion.....187

6. SUMMARY AND PROJECTIONS FOR FURTHER RESEARCH.....196
6.1 Highlights of Issues Addressed in the Thesis.....196
6.2 Projections for Further Research.....199

REFERRENCES.....201

LIST OF TABLES

Table.....Page

1.1 Comparison of Dynamics Formulations for a 6-joint robot ...20

3.1 Dimensions of Planar Manipulator Used in Mode shape Calculations.....116

3.2 First Modal Frequency for Different Masses of Link 2.....117

3.3 First Modal Frequency for Different Joint Configuration...119

4.1 Link Parameters of Example Arm.....134

4.2 Open-loop poles of the flexible subsystem.....149

4.3 Closed-loop poles of the flexible subsystem under Pseudolink control.....149

5.1 Summary of Inverse Dynamics Algorithm for Flexible Manipulators with Generalized accelerations omitted.....174

5.2 Summary of the Linear Recursive Algorithm for Calculating the Inertia Matrix H.....188

LIST OF FIGURES

Figure.....Page

1.1 The PUMA 560 industrial robot.....2

1.2(a) Definition of kinematic parameters of a chain of actuated, rigid links.....10

1.2(b) Coordinate frames of the Stanford manipulator.....10

1.3 Definition of coordinate frame transformation.....16

2.1 Definition of vectors used in the kinematic energy computation.....36

2.2 Example manipulator used in simulation.....66

2.3 Simulation results of the two-link manipulator.....68

3.1 Element definitions for finite element model.....80

3.2 Definition of vectors used in kinetic energy computation...88

3.3 Results of the finite element simulation of the two-link manipulator shown in Figure 2.2.....108

3.4 First mode shape of the first link of a two-link manipulator for varying masses of the second link.....118

3.5 First mode shape of the first link of a two-link manipulator for several values of the second joint angle.....120

3.6 Comparison between link displacement curves obtained by the two different modeling methods.....121

4.1 Block diagram of the computed torque controller.....125

4.2 Three-link manipulator used for simulations.....133

4.3 Joint angle responses when the decoupling signal only is used for control.....135

4.4 Tip deflections of the flexible links of the three-link manipulator under decoupling control only.....137

4.5 Definition of the pseudolink angles.....142

4.6 Joint-angle trajectory prototype for all three joints.....151

4.7 Simulation results showing the pseudolink-angle trajectory of the second link, compared with the desired trajectory.....153

4.8 Simulation results showing the required torques for a flexible manipulator under pseudolink control, compared with the required torques for rigid manipulator control.....155

4.9 Block diagram of the Multirate Controller.....160

4.10 Single-link manipulator used in the multirate example.....161

4.11 Responses of joint angle and flexibility variable under joint controller only. Joint controller operating at 50Hz.....162

4.12 Responses of the joint angle and flexibility variable under multirate control. Joint controller operating at 20Hz, Flexibility controller operating at 100 Hz.....164

5.1 Diagram showing the recursive doubling comptational algorithm.....171

CHAPTER 1. OVERVIEW OF THE RESEARCH PROJECT.

1.1 Introduction.

The problems of modeling and control of robot manipulators that possess significant structural flexibility have recently aroused considerable research interest [1-11]. Traditional manipulator designs have deliberately avoided these problems by specifying relatively massive link-construction. In such designs, the ratio of the length of each link to its thickness (aspect ratio) is decreased to the point where transverse link vibrations are of very small amplitudes and very high frequencies. Such links can be considered as being rigid for all practical purposes. The small link vibrations can then be filtered out of all controller measurements to prevent aliasing in the sampling operation performed by the digital controller. The design of control systems for such manipulator arms, which is a difficult task in itself due to the arm's complex, non-linear dynamics, can therefore avoid the additional complexities that link-flexibility would create. A price is paid, however, in larger actuator requirements in order to move the heavier links. Also, the actuators themselves add considerable mass to be moved if they are physically located at the joints.

Figure 1-1 shows a typical robot manipulator, the Puma 560. This manipulator has rigid links. It has a payload capacity of 2.5 kilograms and a total link mass of 55 kilograms, giving a mass-to-payload ratio of 22 to 1. A large part of the control effort thus goes towards moving the links themselves.

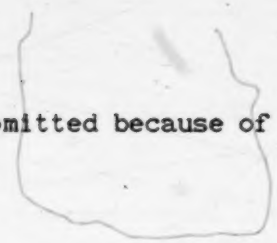


Figure 1.1 omitted because of copyright protection.

Figure 1.1. The PUMA 560 Industrial Robot.
(Taken from Reference 99)

Increasing demands for lower-priced arms that can move at high speeds, yet maintain relatively accurate trajectory-following characteristics, are compelling robot designers to reject the traditional solution provided by the low aspect ratio link designs, and consider new lightweight arm design methods. In such designs, the aspect ratio of each long link is increased. Transverse vibrations due to link flexibility now become significant and can no longer be ignored in control system design for this type of arm. The control system design problem is rendered more difficult by this step but does not become entirely intractable. Much research, however, is needed to solve this

difficult problem.

Flexible manipulator arms offer several advantages other than increased speed and lower overall cost. Some of these are [2]:

1) Lower energy consumption,
2) Safer operation due to reduced inertia,
3) Greater static compliance when in contact with external objects,
4) Greater back-driveability due to elimination of bulky gearing mechanisms, and
5) Lower mass to be transported (especially useful for space applications).

The enhanced compliance made possible by flexible links is a very useful feature in assembly-type applications. In addition, flexible arms are likely to be direct driven, hence advantage (4) above enhances compliance capabilities as well. Flexible arms are sometimes referred to as "compliant arms" to emphasize these characteristics.

The design of stable dynamic control systems for flexible manipulators involves modeling both rigidity and flexibility effects in a suitable manner. These models then form part of the control system design process. Many procedures for modeling flexible manipulators have been proposed in the literature[2,3,12-17]. Some of these are suitable for application to multi-linked arms but by far the majority of them can only be applied to a manipulator with just one or two links, and that too with motion restricted to a plane. In practice, however, a manipulator would be expected to move in three-dimensional space and would likely have several flexible links. Models for such an

arm need to be developed. These models must be computationally efficient since fast and efficient computer simulation is also part of the overall design procedure for both the mechanical and control systems of the manipulator and dynamic models that can facilitate this are certainly needed. Also, calculation of some of the dynamic terms may form part of the low-level controller and must therefore be calculated in real-time.

1.2 Organization of the Thesis.

It is in the area of fast and efficient modeling and control of general, multi-linked flexible arms that this thesis finds its *raison d'etre*. Fast computational schemes for calculating inverse dynamics and inertia matrices of multi-linked, flexible arms are derived in this thesis, and control systems that use them are proposed. Chapters 2 and 3 describe the derivations of the models and the computational algorithms for their synthesis. Both modal and finite element approaches are used in the derivations. The advantages and disadvantages of each approach are discussed.

Chapter 4 describes certain control system strategies. Their performance is evaluated by computer simulation. The main control system strategy involves the design of a composite controller consisting of a two parts. One part controls the gross joint motion and the other part actively damps out the higher frequency vibrations due to flexibility.

In Chapter 5, the recursive properties of the dynamics algorithms are discussed. Similarities between these algorithms and Newton-Euler dynamics algorithms for rigid manipulators are demonstrated. It is shown that efficient parallel implementation

of these algorithms is feasible, and that pipelining methods similar to those employed in parallel implementation of the rigid dynamics algorithms are also applicable to the algorithms presented in the thesis. Chapter 6 is a summary of the thesis and includes proposals for future research.

1.3. Background Material.

In order to provide an adequate perspective for this research, more information on modeling and control of manipulators in general, and of flexible manipulators in particular, must be provided. This is done in the following sections of this chapter. Substantial background material is provided on topics that impinge directly on the main aspects of the thesis. The information, however, is by no means exhaustive and the references at the end should be consulted for a more thorough treatment of these and other topics.

1.4. Justification of Flexibility.

It was stated earlier that the problems associated with controlling a flexible manipulator are very difficult to overcome and require much research effort for their satisfactory solution. More precise justification should therefore be given for engaging in this type of research. In this section, the results of Book and Majette [2] on this topic are summarized. In so doing, the constraints on allowable flexibility are highlighted.

The sizing of a manipulator link in the context of determining desired degree of flexibility is constrained by two main factors:

1) Required strength (maximum stress capability) of the link,

and

2) Stiffness requirement imposed by control system stability and related considerations.

For a link that is in the shape of a long, uniform cylindrical beam, its strength is determined by its radius and by the material from which it is made. If the material is known, only the radius needs to be determined or specified. The maximum stress, σ_{max}, in a uniform cylindrical link is given by:

$$\sigma_{max} = \frac{4 M_{max} R}{\pi R^4} \qquad (1\text{-}1)$$

where R is the radius of the link and M_{max} is the maximum bending moment exerted on the link. In a given application, the maximum stress and the maximum bending moment that will be required are known. Hence the minimum radius, $R_{strength}$, that is required to provide the desired link strength can be calculated.

For adequate strength, we must have:

$$R > R_{strength}. \qquad (1\text{-}2)$$

The stiffness constraint can also be described in terms of a minimum radius $R_{stiffness}$. The stiffness necessary to assure proper control system performance depends, of course, on the control algorithm. To give an idea of the relationship between control system performance and link-stiffness, we cite the results of Book et al [12]. They found that when a single-link manipulator is controlled by a simple PID controller, satisfactory damping

cannot be achieved if the clamped-actuator natural frequency of the link exceeds approximately three times the closed-loop bandwidth of the controlled system. There are several other factors that affect this relationship, but in terms of link-flexibility, the constraint can be expressed as follows:

$$R > R_{stiffness} \qquad (1\text{-}3)$$

If $R_{stiffness} > R_{strength}$, inclusion of flexibility considerations in the control system design can perhaps allow the link's radius to be reduced to the lower limit of $R_{stiffness}$. This reduction in radius results in lowering the speed penalty imposed by actuator size. Faster motion can therefore be realized with the same actuator. An attractive alternative design is that in which the speed is not significantly increased but instead, actuator size is considerably reduced.

The studies reported by Book et al [2] have demonstrated that for arms with high aspect ratios and carrying light payloads, the assumption that $R_{stiffness} > R_{strength}$ is justified. In addition, the following interesting observations are also made:

1) A substantial penalty is paid in terms of speed reduction for stiffening an arm by increasing its radius. Values of 50% have been noted.
2) The penalty is greatest for longer arms and lighter payload.
3) The penalty is greater for relatively high-strength materials with low rigidity and high density.

In this thesis, the major consideration is the desire to

reduce actuator size while maintaining manipulator speed and reasonable end-point accuracy over the gross motion trajectory. We concentrate our efforts on manipulators with links of normal length (about one meter) but with high aspect ratios (i.e. small thicknesses). It is shown in chapter 4 that it is possible to design control systems that can accomodate significant flexibility of the links without sacrificing speed and reasonable accuracy using relatively low powered actuators.

1.5 Modeling of Robot Manipulators.

In this section we examine the fundamentals of kinematic and dynamic modeling of both rigid and flexible manipulators and point out their similarities and differences. Kinematics refers to geometrical and time based properties of motion. The relationships between thses motions and the forces and torques that cause them constitute dynamics.

1.5.1 Kinematics of Rigid Manipulators.

Accurate, direct measurement of the end-point position and velocity of a robot arm in the way humans use vision and feeling is generally not possible. The basis for advanced control of rigid manipulators, then, is a relationship between the Cartesian coordinates of the end-effector and the manipulator joint coordinates. Positioning of the gripper is normally specified in Cartesian, task-oriented coordinates, but is accomplished by actuation of individual joints, in joint coordinates. We therefore need to transform quantities back and forth between the two coordinate systems. This requires consideration of the kinematics

of the manipulator.

A conventional rigid-link manipulator, such as the Stanford Manipulator shown in Figure 1-2(b), consists of a sequence of rigid links connected together by actuated joints. There is considerable theory surrounding its analysis [18,19], but we will consider only certain pertinent areas here. The kinematics of the manipulator arm are simplified somewhat by restricting the motion of each joint to one degree of freedom only, either translational or rotational. Denavit and Hartenberg [20] proposed, in 1955, a system of notation based on matrices for mathematically describing the configuration of kinematic chains, and this notation has since been universally adopted. Basically, this notation involves defining a set of coordinates for each link and a transformation matrix between any pair of coordinate sets. The description given below follows after the modified Denavit-Hartenberg rules given by Paul [19].

Figure 1.2(a) shows the link coordinate system and the four parameters which describe a link with rotational joints at both ends. It shows coordinate system (x_n, y_n, z_n) attached to the n'^{th} link and moving with it. Rotation of the link occurs about the axis of the previous joint. For a rotational joint, the z axis aligns with the n+1'st joint axis.

The four parameters describing the n'^{th} link are the "link length", a_n, the link "twist", α_n, the relative position of two connected links (offset), d_n, and the rotation of link n with respect to the previous link, θ_n. For a rotational joint, all the link parameters are fixed except θ_n, which is the joint variable. For a translational joint, the joint variable is the distance d_n.

Figure 1.2(a) omitted because of copyright protection.

Figure 1.2(a) Definition of Kinematic Parameters of a Chain of Actuated, rigid links. (Taken from Paul[19])

Figure 1.2(b) omitted because of copyright protection.

Figure 1.2(b). Coordinate Frames of the Stanford Manipulator. (Taken from Paul[19])

The manipulator arm is usually attached to a fixed base. Reference coordinates are then taken as base coordinates, (x_0, y_0, z_0), located at joint 1. Figure 1.2(b) shows the alignment of link-coordinates for a six-joint manipulator. There are seven links including the base, and seven coordinate frames, with frame (x_6, y_6, z_6) located at the hand (link 6, since the base is link 0). Once the preceeding system of coordinate frames has been established, the transformation from coordinate frame (x_i, y_i, z_i) to coordinate frame $(x_{i-1}, y_{i-1}, z_{i-1})$ can be represented by the following 4x4 homogeneous transformation matrix:

$$ {}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\alpha_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1\text{-}4) $$

Note that for a translational joint, $a_i = 0$.

By using homogenous transformation matrices one can specify the position and orientation of the end effector in base coordinates by successive application of the A-transformation. The result for an N-link manipulator is a matrix T_N, where:

$$ T_N = {}^0A_1 \, {}^1A_2 \, \text{---} \, {}^{N-1}A_N \qquad (1\text{-}5) $$

If base coordinates do not coincide with workspace (cartesian) coordinates then T_N can be pre-multiplied by the 4x4 matrix which transforms base coordinates into workspace coordinates. The result

is a matrix representing the nonlinear function expressing the position and orientation of the gripper, in workspace coordinates, in terms of the generalized joint variables, q_i. References [18,19] provide a fairly comprehensive treatment of manipulator kinematics as used in robot control and the reader is referred to them for a more thorough treatment of the subject.

The matrix T_N, which provides a way to obtain the Cartesian coordinates of the end-point when the joint angles are given, can readily be obtained for any manipulator and allows calculation of the end-effector position and orientation if the joint angles are known. The inverse of this operation, that is, a method of finding the joint angles that correspond to a given set of Cartesian, end-point, coordinates, is of particular interest in controlling the manipulator. Positioning of the joints to obtain the desired gripper movement in workspace coordinates can then be performed dynamically. However, this "inverse kinematic solution" of manipulator arms with more than two or three joints is sometimes difficult to obtain. This calculation is normally to be embedded in a real-time servo loop and only a very minimum number of mathematical operations may be performed. There are methods of obtaining the inverse kinematic solution for specific manipulators [19,21-25] but they are not necessarily extensible to all manipulators. Those schemes that have proven successful when applied to popular robots have recently seen the postulation of highly parallel computer architectures for their implementation using VLSI technology [26,27]. This approach has drastically reduced computation times of both forward and inverse kinematics and real-time kinematic control is now a reality.

During the period when real-time inverse kinematic computation was impractical, many proposals for advanced manipulator control avoided this problem by controlling gripper rates instead of gripper positions. The technique of controlling gripper rates in cartesian space is referred to in the literature as "resolved motion rate control". This involves computing the Jacobian which can usually be inverted with relative ease, although there is sometimes a singularity problem associated with its inversion. Many numerical techniques exist for computation of the Jacobian [28-30]. Several "resolved" schemes were proposed [31-35] but it is doubtful that they would now be favoured over inverse kinematic schemes, given the removal of computation bottle-necks in the latter schemes. The introduction of link flexibility has again made inverse kinematic solution a monumental task and it is not unlikely that we will see a re-appearence of these resolved schemes.

1.5.2 Kinematics of Flexible Links.

Link flexibility introduces the need for additional kinematic parameters above and beyond those specified for rigid links. These parameters are required to describe the instantaneous deflections of the link at a given point along its axis. Many of the concepts described below are applicable to arbitrarily shaped links but since, as mentioned earlier, the benefits of flexibility are greatest when the links have high aspect ratios, we shall restrict our discussion to manipulators with this configuration. For slender links, the Euler-Bernoulli theory of bending of beams [36] is appropriate. This theory leads to a set of partial differential

equations of the following form:

$$-\frac{\partial^2}{\partial x^2}\left[EI(x)\ \frac{\partial^2 y(x,t)}{\partial x^2}\right] + = m(x)\ \frac{\partial^2 y(x,t)}{\partial t^2} \qquad (1\text{-}6)$$

where:

y(x,t) is the link's deflection from its normal "rigid" position at a distance x form its proximal end, at time t,

EI(x) is the link's flexural rigidity,

m(x) is the differential mass at point x, and

A similar expression describes deflections in the x-z plane. Torsion about the longitudinal x-direction is represented by a similar expression but is of second order.

In general, these equations cannot be solved analytically except in a few special cases, for example simply supported beams. Some sort of discretization is necessary and two methods have been widely used for this purpose, namely, the assumed-modes method [4,5,8,36] and the finite element method[1,6,9,37,38]. A comparative evaluation of the merits of each method is left for Chapter 3. In this section, only a brief discription of the methods in so far as they are relevant to kinematics is given.

In the assumed-modes method, link-deflections and torsional rotations are represented by superposition of products of a finite number of assumed mode-shapes and time-dependent, generalized coordinates. The mode-shapes are often taken as the solution of the partial differential equation (1-6) given simple boundary

conditions such as clamped-clamped, clamped-free or pinned-free. These mode-shapes represent additional kinematic parameters that are needed to adequately describe coordinate transformations.

In the finite element method, each link is divided into a number of elements with interpolating polynomials describing element deflections at interior points. For slender links, these elements are one-dimensional and the interpolating polynomials are often of third order (Hermitian polynomials) [38]. The rotation and translation of the end-point of the last element in the link is used in coordinate transformations.

Let ϕ_{yi} and ϕ_{zi} be rotational angles due to link-deflections of the i^{th} link in the transverse y and z directions respectively, and ϕ_{xi} the angle of torsion about the longitudinal x-axis. Also let Δ_{yi} and Δ_{zi} be displacements of the link's end-point in the y and z directions respectively. A homogeneous transformation matrix, E_i^{rf}, that transforms the "rigid" coordinate system (x_i', y_i', z_i') to the end-point coordinate system (x_i, y_i, z_i), can be defined as shown in Figure 1.3. That is:

$$E_i^{rf} = \text{Rot}(x, \phi_{xi})\,\text{Rot}(y, \phi_{yi})\,\text{Rot}(z, \phi_{zi}) * \text{Trans}(y, \Delta_{zi})\,\text{Trans}(y, \Delta_{xi})\,\text{Trans}(y, \Delta_{yi}) \qquad (1\text{-}7)$$

In the assumed-modes method, the displacements and angles are given by the following expressions:

$$\phi_{xi} = \sum_k \delta_{xik}(t)\,\bar{\phi}_{xik}; \qquad \phi_{yi} = \sum_k \delta_{yik}(t)\,\bar{\phi}_{yik};$$

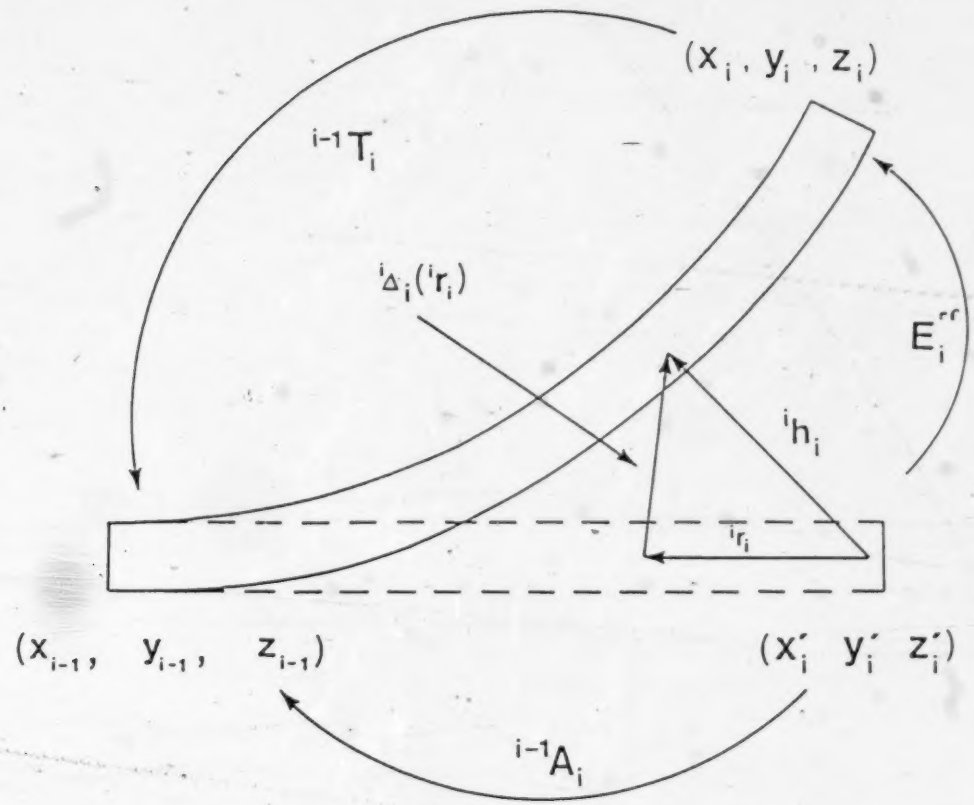


Figure 1.3. Definition of Coordinate Frame Transformations.

$$\phi_{zi} = \sum_k \delta_{zik}(t)\ \bar{\phi}_{zik}\ ; \qquad \Delta_{yi} = \sum_k \delta_{yik}(t)\ \bar{\Delta}_{yik};$$

$$\Delta_{zi} = \sum_k \delta_{zik}(t)\ \bar{\Delta}_{zik}.$$

where $\bar{\phi}_{xik}$, $\bar{\phi}_{yik}$, $\bar{\phi}_{zik}$, $\bar{\Delta}_{yik}$ and $\bar{\Delta}_{zik}$ are the assumed rotation and translation of the k'th mode-shape at the link's tip, and $\delta_{xik}(t)$,

$\delta_{yik}(t)$, and $\delta_{zik}(t)$ are generalized coordinates. In the finite-element method, ϕ_{xi}, ϕ_{yi}, ϕ_{zi}, Δ_{yi} and Δ_{zi} are the generalized coordinates themselves. In a simulation, these would be generated successively by numerical integration. In a control system, it would be convenient if these quantities can be measured.

The transformation between base and end-effector is then given by cascading the joint and flexibility transformation matrices to form T_N. That is:

$$T_N = {}^0A_1 \, E_1^{rf} \, {}^1A_2 \, E_2^{rf} \, \text{---} \, {}^{N-1}A_N \, E_N^{rf} \qquad (1\text{-}10)$$

1.6 Manipulator Dynamics.

Manipulator dynamic models play a crucial role in the design of robot control systems. Various approaches are available to formulate the dynamics of rigid robot arms, such as the Lagrange-Euler [19,39-41], Newton-Euler [42-44], recursive Lagrange-Euler [41], and Kane's equations[45,46]. The Lagrange-Euler equations are well structured and can be expressed in matrix notation, as follows:

$$\underset{\sim}{\tau} = H(\underset{\sim}{q})\ddot{\underset{\sim}{q}} + \underset{\sim}{h}(\underset{\sim}{q}, \dot{\underset{\sim}{q}}) \qquad (1\text{-}11)$$

where:

$\underset{\sim}{\tau}$ is the vector of generalized driving torques at the joints,

$\underset{\sim}{q}$ is the vector of generalized joint positions,

$\underset{\sim}{H}$ is the NxN inertia matrix that includes effects from both the actuator system and manipulator links,

$\underline{h}$ is a vector of centrifugal, coriolis, viscous frictional and gravitational generalized forces, and

N is the number of degrees of freedom (joints) of the manipulator.

These equations are highly nonlinear and tightly coupled. The magnitudes of the elements of the matrix H can vary widely in a nonlinear manner with arm configuration. Gravitational effects are a maximum when the arm is outstretched and horizontal, and zero when pointing downwards. Coriolis and centrifugal effects are small when the arm is moving slowly but cannot be neglected when fast, fairly accurate trajectory following is being performed. Joint actuator non-linearities, such as static friction, gear backlash and magnetic saturation effects are not included in the above equation. Some robot calibration schemes [47,48] have attempted to measure these effects and compensate for them in the control system but this is not a major concern in this thesis. In fact, actuator dynamics are not considered at all. Instead, we will concentrate on the dynamic effects generated by arm motion since these form the major obstacles to manipulator control. A reasonable model of actuator systems is given in reference [49] and it is shown there how to incorporate it into arm models. It can be incorporated in the models developed in this thesis in exactly the same way.

The form of these equations can be exploited in designing manipulator control systems but the original Lagrange-Euler formulation is computationally much too intensive to implement in real-time. Much work has recently been done in order to reduce the computational burden of these equations. Paul [39] and Beczsy [40]

neglected certain terms, notably the coriolis and centrifugal terms, in their manipulator studies and observed significant reductions in computational complexity but with reduced accuracy. Raibert and Horn [50] used a table look-up approach. The disadvantage is the requirement of a large memory for sustained accuracy. Hollerbach [41] exploited the recursive nature of the equations and significantly reduced the computational time without sacrificing accuracy. The structure of the equations was destroyed, however, making them less useful for control system design purposes. This disadvantage was overcome by using Hollerbach's more efficient method for symbolic generation of closed-form equations [51,52].

The computational complexity of the Newton-Euler formulation [43] depends linearly on the number of manipulator links and this modeling approach is even more efficient than the recursive Lagrangian formulation. However, it lacks the matrix structure of the Lagrange-Euler equations. Real-time implementations of the manipulator dynamic equations are required as part of some advanced control system strategies, such as "computed-torque" control [40] and the resolved schemes. Because of its numerical efficiency, the Newton-Euler algorithm is suitable for computing these dynamic terms.

Forward Newton-Euler algorithms have also been developed. [44]. In addition, the method has been used for symbolic generation of closed-form algebraic equations [53,54]. An interesting development has been the appearance of several parallel-processing schemes for implementation of the Newton-Euler equations [55-62]. This has been made possible because these

Table 1.1

Comparison of Dynamics Formulations for 6-joint robot (Adapted from Hollerbach [41] and Lathrop [61]

Method	Mults	Adds.
Original Lagrangian	66,271	51,548
Recursive Lagrangian	2,195	1,719
Newton-Euler	852	738
Linear parallel Newton-Euler (Lathrop)	15	43
Logarithmic parallel Newton-Euler (Lathrop)	11	2
Systolic Pipeline (Lathrop)	1	3

equations can be organized into linear recurrence form [62]. As a result, the time required for their computation has been reduced considerably and real-time computation is now a real possibility.

Kane's equations are also recursive and are actually more efficient than the Newton-Euler equations, but they do not possess a linear recursive form and as such have not been widely studied in the literature. Uniprocessor implementation is fast but not fast enough for real-time computation. Parallel computation of the Newton-Euler equations is therefore more attractive for control system implementation.

Table 1.1 gives a comparison of the complexity of the different formulations of the dynamic equations, including parallel implementations of the Newton-Euler equations, for rotary 6-link manipulators.

1.7 Dynamics of Flexible Manipulators.

Consideration of flexibility in arm dynamic models started in the early 1970's [12]. In some of these works, manipulators which move in 3-dimensional space (spatial arms) were considered. Link-flexibility, which in reality is distributed along the length of the link, was represented in the time domain in terms of lumped masses and springs. Book et al [12], however, modeled distributed link-flexibility in the frequency domain via transfer matrices. Most of these models have been essentially linear and have not been generally extensible to spatial arms with more than two links.

Researchers whose works have previously been in the field of modeling and control of flexible spacecraft have also applied their ideas to modeling and control of flexible manipulators as well. The works of Hughes [70] and Singh and Likins [71], are worth noting. Both linear and nonlinear models were developed in this research. The nonlinear models were accurate but required tremendous amounts of computer time for simulation. Linearized models ignored some coupling effects between rigid and flexible dynamics and while this might be appropriate for the slow-moving space shuttle arm, it is not for more general, fast moving, industrial arms.

More recent work addressed specifically to modeling flexible manipulator arms has been done by Sunada and Dubrowsky [1], Usoro [6], Book [4] and Hastings[8]. Sunada and Dubrowsky developed an impressive finite-element procedure for determining dynamic deflections in a flexible link with an arbitrary shape. Their model, however, was linearized about a nominal trajectory and

hence would not provide accurate simulation. A Lagrangian approach is employed, utilizing 4x4 homogeneous transformation matrices. Usoro also utilized 4x4 matrices in a Lagrangian, finite-element formulation. Although the formulation was derived only for a two-link, planar manipulator, the principles are extensible to spatial, multi-link flexible arms. The use of 4x4 transformation matrices is, of course, inefficient and this may inhibit its use in a simulation where speed is a factor.

Book [4] generated the complete, nonlinear dynamics using the Lagrangian approach and 4x4 transformation matrices. Unlike Usoro's method, Book represented kinematics by a modal expansion. This results in fewer degrees of freedom for comparable accuracy.

The works of Usoro and Book illustrate the two main approaches that are being taken by current researchers to model flexible manipulators, namely, the Lagrangian-Finite-Element method and the Lagrangian-Assumed-Modes method. In the assumed-modes method, link-deflections are represented by the superposition of products of mode shapes and time-dependent generalized coordinates. Simple boundary conditions are assumed at the joints and these are used to derive analytic expressions for the mode-shapes.

There is some uncertainty over whether these mode shapes represent the actual vibration mode shapes with sufficient accuracy. Furthermore, the correspondence between the generalized coordinates as defined in the assumed-modes method and physical deflection measurements, is not straightforward. In the finite-element method, these problems do not exist. The actual boundary conditions can be more accurately represented in the finite-element equations. In addition, the generalized coordinates

are now actual deflections of the end-points of elements. These deflections can conceivably be measured by strain gauges fixed at element boundaries.

The elegance of the finite-element method, however, is not without a price. This method is noted for requiring long CPU times for problem solution. Two factors contribute to these high computational requirements. Firstly, the Lagrangian approach that is used in most finite-element programs to calculate dynamic terms leads to numerical algorithms that are inefficient [41]. Secondly, the finite-element method itself results in a very high model dimension. For example, if the actual vibration mode-shapes were known, then four modes of vibration may be sufficient to describe link deflections in a flexible manipulator using the assumed-modes method. The finite element method may require ten or more elements, resulting in fifty or more degrees of freedom per link. Computational speed and efficiency are needed in order to quickly simulate alternative manipulator mechanical designs for the flexible manipulator. Even greater speed is required for real-time implementation of the control system since feed-forward terms have to be calculated in order to achieve decoupling of flexibility effects from rigid-body dynamics [72].

1.8. Contributions of the Thesis to Flexibile Manipulator Modeling.

This thesis addresses the problem of algorithmic inefficiency of the traditional Lagrangian approach to both the assumed-modes and finite element methods.

A significant development in robot dynamics is the proof by

Silver [63] that the differences in computational complexity between the different formulations arise from three major sources:

(1) the way in which manipulator kinematic transformations are represented in the methods,

(2) the way angular velocities are represented, and

(3) recursiveness.

The traditional Lagrangian method is not recursive and utilizes 4x4 homogeneous transformations which tend to be inefficient. The Newton-Euler method separates rotational transformations from translations and uses 3x3 matrices to represent the former whilst position vectors are used to represent the latter. Hollerbach's recursive Lagrangian formulation also does the separation between rotations and translations and this, together with its recursiveness, accounts for its relative speed. The Newton-Euler method, however, goes one step further and represents angular velocities in terms of moving coordinate systems and 3-element vectors. The Lagrangian methods continue to use transformation matrices and their derivatives and are therefore slower. Silver shows that the Newton-Euler equations for rigid manipulators can actually by derived from the Lagrangian equations if the above changes are made.

The basic Newton-Euler equations of motion require insight into the motions of the physical system to which they are being applied and are difficult to apply to a flexible manipulator. Use of the Lagrangian equations is much more straightforward. The implication of Silver's proof of equivalence is that one does not necessarily need to pursue the direct Newton-Euler approach in order to gain the advantages of the traditional Newton-Euler

formulation. The modified Lagrangian approach can be taken instead and this is exactly what is done in this thesis in developing fast, efficient and accurate models of flexible manipulator dynamics for both control system implementation and simulation of the physical system. In Chapter 2, this technique is applied to develop fast and efficient recursive algorithms for calculating both inverse and forward dynamics of flexible manipulators, using the assumed-modes method. Similar finite-element algorithms are developed in Chapter 3. These algorithms are quite fast and are easily written in the form of linear recurrences suitable for parallel implementation.

The uncertainty over whether assumed mode shapes describe link-deflections with sufficient accuracy is also addressed in Chapter 3. A comparison is made between simulation results obtained from both the assumed modes and finite element methods. The results from both methods are remarkably similar. The finite element method takes the changing boundary conditions into account in a more accurate way but since the assumed modes method yields similar results to the finite element method, the assumed modes method seems to model the flexible dynamics with sufficient accuracy.

1.9 Modern Approaches to Manipulator Dynamic Control.

1.9.1. Rigid Manipulators.

A large number of proposals for advanced control of rigid-linked robots have recently been published in the open literature. These proposals take into account, in one way or

another, the complex, nonlinear and tightly coupled robot dynamics. One approach which has been taken by many authors [33,39,40,74-77], is to cancel the nonlinear dynamics by nonlinear feedback. The resulting closed-loop dynamics are decoupled and linear, hence linear design methods can then be used to place the closed-loop poles at desired locations. These methods have been termed "Computed Torque" methods. The early computed torque methods of Markiewicz [74] and Bejczy [40] suffered from the disadvantage that the manipulator dynamics had to be known precisely, hence the method was not robust. Some calibration schemes attempted to solve this problem by on-line calculation of payload and inertia parameters and hence to satisfy the assumption that the dynamics are known. Other approaches to improving the robustness of computed torque methods have added an integral term or an acceleration feedback loop to the existing position and velocity loops [80,81]. In the latter schemes, robust behavior has been achieved in a relatively simple manner. Another disadvantage of the computed torque schemes has been the high computational requirements but this is no longer a problem, as we have already mentioned.

The other approach to advanced robot control system design is the adaptive one. One of the earliest adaptive proposals was the model-referenced adaptive controller of Dubrowsky and DesForges [82] in which the feedback gains are modified on-line in order for the closed-loop dynamics to follow a decoupled, linear reference model. Later designs include self-tuning control [83] and resolved motion adaptive control [35]. These strategies are robust but are difficult to analyse. They can also be computationally intensive

but this is less of a problem in this age of fast signal processing chips.

There are several other schemes which optimize various quantities such as time along the trajectory [84], or energy [85], but again they are difficult to analyse.

1.9.2. Flexible Manipulators.

It has already been mentioned in section 1.2 that when a flexible manipulator is controlled by simple feedback of joint positions and velocities, system stability is lost when the feedback gains are high [12]. This limitation prevents achieving the speed requirements that are needed to make flexible manipulators attractive. Other researchers [85] have also found that when more advanced control schemes based on rigid dynamics only are applied directly to the flexible manipulator system, stability suffers when desired speed in increased. The flexible dynamics need to be included in the control system model.

Early control systems of single or double linked, flexible arms have been based on linearized dynamics [3,5,86]. Optimal or PID controllers have been synthesized and good performance have been noted. The experiments by Cannon and Schmitz [1], in particular, have received widespread fame. They have highlighted the problems due to non-colocation of actuators and sensors and have pointed out that when this situation occurs, as it does in flexible manipulator control, an accurate model is necessary for good response. They have also discussed the problems associated with inadequately representing distributed link flexibility in the model. The effects of the so called "observation spillover" and

"control spillover" were mentioned but better treatments of these effects have been given by Balas [87] and Truckenbrodt [88]. These problems and effects will be more fully discussed in later chapters.

Linear control systems for flexible manipulators are adequate for single or double link arms moving in a plane, and over short distances. In Cannon and Schmitz's experiments, the tip of their single link was given a commanded displacement of only six centimeters. The maximum joint angle deflection was only six degrees. Faster motions over wider trajectories are unlikely to follow the commanded trajectories if the controllers are based on simple, linear models.

Adaptive control strategies have recently been proposed [98,90]. These are based on the assumption of accurate, direct sensing of end-point position and velocities in Cartesian space. This assumption is not quite valid since fast Cartesian sensors are not yet available. It is easier to measure the relative position and velocity of the tip of each link with respect to the previous joint. This allows kinematic transformations in both the forward and inverse directions as in rigid manipulators. This solves the non-colocation problem and servoing in the joint space is again sufficient. This approach is taken in Chapter 4 and more details of the method are given there.

The emphasis in this thesis is not on control system design but on fast algorithms and parallel implementations. We need to show, however, why fast algorithms and parallel processing are needed. It is for this reason that the thesis includes a chapter on control system design. Computed torque techniques for flexible

manipulator control are developed in chapter 4. The main advantage of these methods is that designing them is quite systematic and straightforward. The methods are easily extensible to spatial, multi-linked flexible arms. Manipulator stability and speed are both maintained.

Computed torque methods require on-line computation of dynamic terms and parallel Newton-Euler algorithms have been successful in achieving the required sampling time for control of rigid arms. In chapter 5, we show how the methods for parallelizing the rigid Newton-Euler algorithm can be directly extended to the parallelization of the "Newton-Euler-like" algorithms of Chapter 2.

CHAPTER 2. RECURSIVE ASSUMED-MODES MODELING OF FLEXIBLE MANIPULATORS

2.1 Introduction.

It is well documented [41] that the Lagrangian approach to manipulator modeling leads to differential equations that are quite intensive from a computational standpoint. In this respect, the Newton-Euler approach has proved very successful in modeling rigid manipulators. As mentioned in Chapter 1, application of the direct Newton-Euler approach to a chain of flexible bodies, each experiencing bending, torsional and compressional vibrations, is not straightforward. In this chapter, this difficulty is circumvented by maintaining the Lagrangian approach but representing rotational kinematics by angular velocities, and coordinate transformations by 3x3 rotational matrices and position vectors. The theory of moving coordinate systems is applied as in the traditional Newton-Euler approach to derive recursive expressions for velocities and accelerations of the coordinate frames. The Lagrangian equations are then applied. The final result of this procedure is a set of recursive equations that are very similar to the Newton-Euler recursive equations [33,44]. A nonlinear flexible model is obtained and the recursive procedure is computationally very efficient. Both inverse and forward dynamics are calculated.

In the inverse plant algorithm, kinematics are calculated on a

forward recursion from the base to the tip and generalized forces are calculated on the return recursion. In the forward dynamics algorithm, a direct recursive algorithm is presented for calculating the terms of the inertia matrix (including flexibility effects). We have chosen the assumed-modes method in this chapter to model the elastic link deflections because of its simplicity. In the next chapter, we present a similar set of recursive algorithms that model link-deflections by the finite-element method.

In the sections that follow, the development of the inverse dynamics algorithm is presented in some detail. Proofs of important assertions are also given. A comparison between the number of computations required for the formulations of this chapter and the number required for Book's Recursive Lagrangian Dynamics formulation [4] is presented in section (2-7). It is shown that the former method offers significant advantages in computational speed without degrading numerical accuracy. In section (2-8), a discussion of the simulation accuracy achieved by the assumed-modes method is presented. Some practical implementational problems in flexible manipulator control are also discussed. Finally, simulation results of an example arm, moving in three dimensions, are given in section (2-9).

2.2 Calculation of Link-Velocities and Accelerations.

The definitions of the link-coordinate frames have already been given in Chapter 1. In what follows, the reader should refer to Figure (1-3). As shown in that diagram, the position of any

point on the i'th link with respect to frame (x',y',z') can be represented by a vector

$$ {}^{i}\underset{\sim}{h}_i = {}^{i}\underset{\sim}{r}_i + {}^{i}\underset{\sim}{\Delta}_i({}^{i}\underset{\sim}{r}_i) \qquad (2\text{-}1) $$

where ${}^{i}\underset{\sim}{r}_i$ is the position vector of the point when the link is undeflected and ${}^{i}\underset{\sim}{\Delta}_i({}^{i}\underset{\sim}{r}_i)$ is the deflection of the point from its undeformed position. (Left superscripts indicate the coordinate system of reference. Right subscripts represent the link number. Wavy-underscored symbols represent 3-element column vectors.) ${}^{i}\underset{\sim}{\Delta}_i({}^{i}\underset{\sim}{r}_i)$ can be represented by a truncated modal expansion, as follows:

$$ {}^{i}\underset{\sim}{\Delta}_i({}^{i}\underset{\sim}{r}_i) = \sum_{k=1}^{m_i} \left[\delta_{ik} \right] \left[0 \mid \Delta_{yik}({}^{i}\underset{\sim}{r}_i) \mid \Delta_{zik}({}^{i}\underset{\sim}{r}_i) \right]^T \qquad (2\text{-}2) $$

where $\left[\delta_{ik} \right] = \mathrm{diag}\left[\delta_{xik}\ \delta_{yik}\ \delta_{zik}\right]$ is the matrix of deflection variables and is multiplied by the mode shapes in (2-2). m_i is the number of modes in the the modal expansion of the i'th link. Note that, as mentioned in Chapter 1, axial vibrations are neglected. We also make the assumption that torsional rotations are small and can be thought of as occuring about the undeflected x-axis.

In the equations that follow, it is useful to define a rotational transformation matrix ${}^{0}T_i^r$ which transforms the "rigid" frame (x',y',z') to base coordinates, as follows:

$$ {}^{0}T_i^r = {}^{0}T_{i-1}\ {}^{i-1}A_i \qquad (2\text{-}3) $$

Recursive expressions for the angular and linear velocities of the coordinate frames (x'_i, y'_i, z'_i) and (x_i, y_i, z_i), referred to the base coordinate frame, can now be written as follows:

$$ {}^{0}\underset{\sim}{\omega}_{i}^{r} = {}^{0}\underset{\sim}{\omega}_{i-1} + \underset{\sim}{z}_{i-1}\dot{q}_i \qquad (2\text{-}4) $$

$$ {}^{0}\underset{\sim}{\omega}_{i} = {}^{0}\underset{\sim}{\omega}_{i}^{r} + {}^{0}\underset{\sim}{\dot{\phi}}_{i} \qquad (2\text{-}5) $$

$$ {}^{0}\underset{\sim}{\dot{\omega}}_{i}^{r} = {}^{0}\underset{\sim}{\dot{\omega}}_{i-1} + \left[{}^{0}\underset{\sim}{\omega}_{i-1} \times \underset{\sim}{z}_{i-1}\dot{q}_i\right] + \underset{\sim}{z}_{i-1}\dot{q}_i \qquad (2\text{-}6) $$

$$ {}^{0}\underset{\sim}{\dot{\omega}}_{i} = {}^{0}\underset{\sim}{\dot{\omega}}_{i}^{r} + {}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{\dot{\phi}}_{i} + {}^{0}\underset{\sim}{\ddot{\phi}}_{i} \qquad (2\text{-}7) $$

$$ {}^{0}\underset{\sim}{v}_{i}^{r} = {}^{0}\underset{\sim}{v}_{i-1} + \left[{}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{p}_{i}^{r^*}\right] \qquad (2\text{-}8) $$

$$ {}^{0}\underset{\sim}{v}_{i} = {}^{0}\underset{\sim}{v}_{i}^{r} + {}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{\Delta}_{i} + {}^{0}\underset{\sim}{\ddot{\Delta}}_{i} \qquad (2\text{-}9) $$

$$ {}^{0}\underset{\sim}{\dot{v}}_{i}^{r} = {}^{0}\underset{\sim}{\dot{v}}_{i-1} + {}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{p}_{i}^{r^*} + {}^{0}\underset{\sim}{\omega}_{i}^{r} \times \left[{}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{p}_{i}^{r^*}\right] \qquad (2\text{-}10) $$

$$ {}^{0}\underset{\sim}{\dot{v}}_{i} = {}^{0}\underset{\sim}{\dot{v}}_{i}^{r} + {}^{0}\underset{\sim}{\dot{\omega}}_{i}^{r} \times {}^{0}\underset{\sim}{\Delta}_{i} + {}^{0}\underset{\sim}{\omega}_{i}^{r} \times \left[\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{\Delta}_{i}\right] $$

$$ + 2\left[{}^{0}\underset{\sim}{\omega}_{i}^{r} \times {}^{0}\underset{\sim}{\Delta}_{i}\right] + {}^{0}\underset{\sim}{\ddot{\Delta}}_{i} \qquad (2\text{-}11) $$

where:

$$ {}^{0}\underset{\sim}{\dot{\phi}}_{i} = \left[{}^{0}\dot{\phi}_{xi} \quad {}^{0}\dot{\phi}_{zi} \quad {}^{0}\dot{\phi}_{yi}\right]^{T} $$

$$ {}^{0}\underset{\sim}{\Delta}_{i} = \left[0 \quad {}^{0}\Delta_{yi} \quad {}^{0}\Delta_{zi}\right]^{T} $$

$^{0}\underset{\sim}{p}_{i}^{r^{\bullet}}$ = vector from origin of $(x_{i-1}, y_{i-1}, z_{i-1})$ to origin of (x'_i, y'_i, z'_i)

$$= \begin{bmatrix} a_i & d_i \sin\alpha_i & d_i \cos\alpha_i \end{bmatrix}^T$$

$$\underset{\sim}{z}_{i-1} = {}^{i-1}A_i \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

($\dot{}$) indicates time differentiation with respect to the primed (rigid) coordinate system of the link.

In the formulation that follows, velocities and accelerations of centers of mass of the elements are also required. These are given by the following equations:

$$^{0\hat{}}\underset{\sim}{\omega}_i^r = {}^{0}\underset{\sim}{\omega}_i^r \tag{2-12}$$

$$^{0\hat{}}\dot{\underset{\sim}{\omega}}_i^r = {}^{0}\dot{\underset{\sim}{\omega}}_i^r \tag{2-13}$$

$$^{0\hat{}}\underset{\sim}{v}_i^r = {}^{0}\underset{\sim}{v}_i^r + \left[{}^{0}\underset{\sim}{\omega}_i^r \times {}^{0\hat{}}\underset{\sim}{s}_i^r \right] \tag{2-14}$$

$$^{0\hat{}}\dot{\underset{\sim}{v}}_i^r = {}^{0}A_{i-1}\left[{}^{i-1}\dot{\underset{\sim}{v}}_{i-1} \right] + {}^{0\hat{}}\underset{\sim}{\omega}_i^r \times \left[{}^{0}\underset{\sim}{p}_i^{r^{\bullet}} + {}^{0\hat{}}\underset{\sim}{s}_i \right]$$

$$+ {}^{0\hat{}}\underset{\sim}{\omega}_i^r \times \left[{}^{0\hat{}}\underset{\sim}{\omega}_i^r \times \left[{}^{0}\underset{\sim}{p}_i^{r^{\bullet}} + {}^{0\hat{}}\underset{\sim}{s}_i \right] \right] \tag{2-15}$$

In the recursive algorithm, these equations would be calculated on the forward recursion from the manipulator base to its tip.

2.3. Formulation of the System Kinetic Energy.

It can be seen from Figure 2-1 that the kinetic energy of link i in deformation is given by:

$$KE_i = \frac{1}{2} \int \dot{\underset{\sim}{r}}_i \cdot \dot{\underset{\sim}{r}}_i \, dm \qquad (2\text{-}16)$$

But notice that:

$$\underset{\sim}{r}_i = \underset{\sim}{R}_i + \underset{\sim}{r}_{i,m} \qquad (2\text{-}17)$$

and

$$\dot{\underset{\sim}{r}}_i = \dot{\underset{\sim}{R}}_i + \left[{}^{0}\underset{\sim}{\omega}_i^{r} \times \underset{\sim}{r}_{i,m} \right] + \mathring{\underset{\sim}{r}}_{i,m} \qquad (2\text{-}18)$$

$$\underset{\sim}{r}_{i,m} = \underset{\sim}{r}^{r}_{i,m} + \underset{\sim}{\Delta}_i(\underset{\sim}{r}^{r}_{i,m}) \qquad (2\text{-}19)$$

$$\mathring{\underset{\sim}{r}}_{i,m} = \mathring{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^{r}_{i,m}) \qquad (2\text{-}20)$$

The symbols $\underset{\sim}{r}_i$, $\underset{\sim}{R}_i$, $\underset{\sim}{r}_{i,m}$, $\underset{\sim}{r}^{r}_{i,m}$ and $\underset{\sim}{\Delta}_i(\underset{\sim}{r}^{r}_{i,m})$ are defined in Figure 2-1. We substitute (2-17) to (2-20) into (2-16), and define the following:

$$\underset{\sim}{\Delta}_{ik}(\underset{\sim}{r}^{r}_{ik}) = \left[\Delta_{xik}(\underset{\sim}{r}^{r}_{i,m}) \mid \Delta_{yik}(\underset{\sim}{r}^{r}_{i,m}) \mid \Delta_{zik}(\underset{\sim}{r}^{r}_{i,m}) \right]^T$$

$$e_{ik} = \int_{\text{link } i} \underset{\sim}{\Delta}_{ik}(\underset{\sim}{r}^{r}_{i,m}) \, dm$$

$$\hat{\underset{\sim}{\Delta}}_i = \sum_{k=1}^{m_i} \left[\delta_{ik} \right] e_{ik}$$

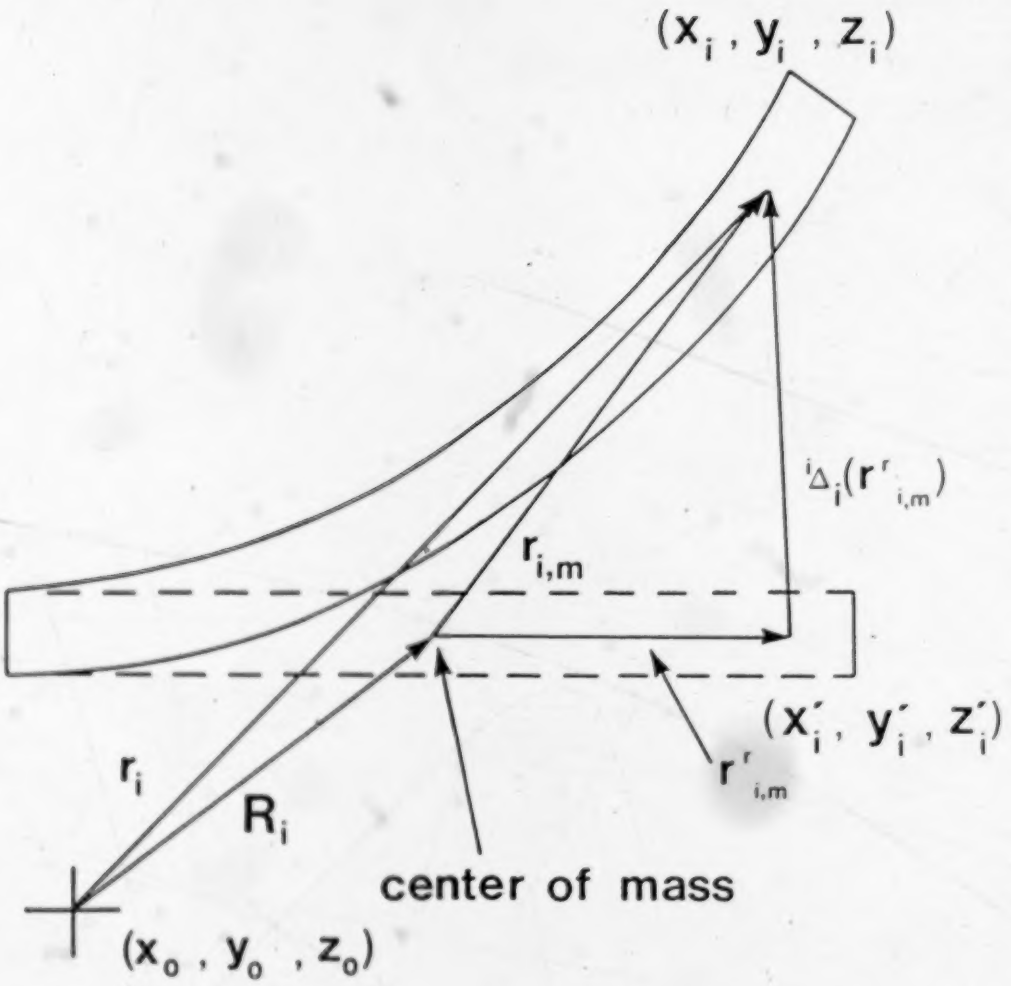


Figure 2.1. Definition of Vectors used in the Kinetic Energy Computation.

$$\overset{\circ}{h}_i = \int_{\text{link } i} \left[\underset{\sim}{r}^r_{i,m} \times \overset{\circ}{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^r_{i,m}) \right] dm$$

$$\overset{\circ}{f}_i = \int_{\text{link } i} \left[\overset{\circ}{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^r_{i,m}) \times \overset{\circ}{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^r_{i,m}) \right] dm$$

$$\overset{\circ}{b}_i = \int_{\text{link } i} \left[\overset{\circ}{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^r_{i,m}) \cdot \overset{\circ}{\underset{\sim}{\Delta}}_i(\underset{\sim}{r}^r_{i,m}) \right] dm$$

$$+ \int_{\text{link } i} \left[\overset{\circ}{\underset{\sim}{\phi}}_{ix}(\underset{\sim}{r}^r_{i,m}) \right] dm$$

Note that $\dot{\underset{\sim}{R}}_i = {}^{0\hat{}r}\underset{\sim}{v}_i$. The kinetic energy of link i, including kinetic energy due to deflection, becomes:

$$KE_i = \frac{1}{2} M_i \left({}^{0\hat{}r}\underset{\sim}{v}_i \cdot {}^{0\hat{}r}\underset{\sim}{v}_i \right) + \frac{1}{2} \left({}^{0\hat{}r}\underset{\sim}{\omega}_i \cdot I_i^f \, {}^{0\hat{}r}\underset{\sim}{\omega}_i \right)$$

$$+ \overset{\circ}{b}_i + {}^{0\hat{}r}\underset{\sim}{v}_i \cdot \left[\left({}^{0\hat{}r}\underset{\sim}{\omega}_i \times \hat{\Delta}_i \right) + \hat{\overset{\circ}{\Delta}}_i \right]$$

$$+ {}^{0\hat{}r}\underset{\sim}{\omega}_i \cdot \left[{}^{0}\overset{\circ}{h}_i + {}^{0}\overset{\circ}{f}_i \right] \qquad (2\text{-}21)$$

The first term of (2-21) is the translational kinetic energy of the link, assuming no deformation. M_i is the link mass. The second term is the rotational kinetic energy assuming the link to be frozen in its deformed state. I_i^f is the 3x3 instantaneous link inertia tensor of the i'th link with respect to its undeformed centre of mass. It is given by:

$$I_i^f = I_i^r + \left[\sum_{k=1}^{m_i} [\delta_{ik}] \left(I_{ik}^{rf} + \left[I_{ik}^{rf} \right]^T \right) + \sum_{k=1}^{m_i} \sum_{l=1}^{m_i} [\delta_{ik}] \, I_{ikl}^{ff} \, [\delta_{il}]^T \right]$$

If we define the following:

$$J_i^r = \int_{link\ i} \left[\underset{\sim}{r}_{i,m}^r \right] \left[\underset{\sim}{r}_{i,m}^r \right]^T dm$$

$$J_{ik}^{rf} = \int_{link\ i} \left[\underset{\sim}{\Delta}_{ik}(\underset{\sim}{r}_{i,m}^r) \right] \left[\underset{\sim}{r}_{i,m}^r \right]^T dm$$

$$J_{ikl}^{ff} = \int_{link\ i} \left[\underset{\sim}{\Delta}_{ik}(\underset{\sim}{r}_{i,m}^r) \right] \left[\underset{\sim}{\Delta}_{ik}(\underset{\sim}{r}_{i,m}^r) \right]^T dm$$

and X is the identity tensor, then:

$$I_i^r = \text{trace}(J_i^r)X - J_i^r$$

$$I_{ik}^{rf} = \text{trace}(J_{ik}^{rf})X - J_{ik}^{rf}$$

$$I_{ikl}^{ff} = \text{trace}(J_{ikl}^{ff})X - J_{ikl}^{ff}$$

The total kinetic energy of the manipulator is the sum of all link kinetic energies:

$$K = \sum_i KE_i$$

2.4. Equations of Motion of the Flexible Manipulator.

2.4.1. Lagrangian Equations of Motion.

Two coupled sets of equations are required to describe the motion of a system of flexible linkages. One set describes the gross movement of the joints and the other describes the vibrations of the links. In these equations, we neglect the gravitational potential energy initially but include its effects later by giving the manipulator base an acceleration equal to that of gravity. The only potential energy term included is the elastic potential energy, P_e. Note that P_e is independent of both q_j and $\dot{q}_j$. If we assume that modal deflections and rotations have no displacements at the points where external forces are applied, then the Lagrangian equations, for the j^{th} joint and the f^{th} flexibility generalized variable, become:

Joint Equation:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{q}_j}\right] - \frac{\partial K}{\partial q_j} = \tau_j \qquad (2\text{-}22)$$

Flexibility Equation:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \delta_{jf*}}\right] - \frac{\partial K}{\partial \delta_{jf*}} + \frac{\partial P_e}{\partial \delta_{jf*}} = \tau_{jf*} \qquad (2\text{-}23)$$

where δ_{jf*} is one of δ_{jfx}, δ_{jfy} or δ_{jfz}. Each of these equations is examined in detail in the following sections.

2.4.2 Joint Equations of Motion.

The Lagrangian equation of motion for joint j requires the derivatives indicated in (2-22). In forming these derivatives, it should be noted that $\overset{\circ}{b}_i$ is a scalar quantity and hence independent of reference frame. It is also independent of $\dot{q}_j$ and hence not part of the joint equations of motion. Note also that I_i^f is symmetric and also independent of $\dot{q}_j$.

After some algebraic simplification, we can write $\frac{\partial K}{\partial \dot{q}_j}$ as follows:

$$\frac{\partial K}{\partial \dot{q}_j} = \sum_i \left\{ \left[\frac{\partial\, {}^{0}\hat{v}_{\sim i}^{r}}{\partial \dot{q}_j} \right] \cdot \left[M_i\, {}^{0}\hat{v}_{\sim i}^{r} + {}^{0}\hat{\omega}_{\sim i}^{r} \times {}^{0}\hat{\Delta}_{\sim i} + {}^{0}\overset{\circ}{\hat{\Delta}}_{\sim i} \right] \right.$$

$$\left. + \left[\frac{\partial\, {}^{0}\hat{\omega}_{\sim i}^{r}}{\partial \dot{q}_j} \right] \cdot \left[I_i^f\, {}^{0}\hat{\omega}_{\sim i}^{r} + \left[{}^{0}\hat{\Delta}_{\sim i} \times {}^{0}\hat{v}_{\sim i}^{r} \right] + {}^{0}\overset{\circ}{h}_{\sim i} + {}^{0}\overset{\circ}{f}_{\sim i} \right] \right\} \quad (2\text{-}24)$$

and:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{q}_j} \right] = \sum_i \left\{ \frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{v}_{\sim i}^{r}}{\partial \dot{q}_j} \right] \cdot \left[M_i\, {}^{0}\hat{v}_{\sim i}^{r} + {}^{0}\hat{\omega}_{\sim i}^{r} \times {}^{0}\hat{\Delta}_{\sim i} + {}^{0}\overset{\circ}{\hat{\Delta}}_{\sim i} \right] \right.$$

$$+ \left[\frac{\partial\, {}^{0}\hat{v}_{\sim i}^{r}}{\partial \dot{q}_j} \right] \cdot \left[M_i\, {}^{0}\hat{\dot{v}}_{\sim i}^{r} + {}^{0}\hat{\dot{\omega}}_{\sim i}^{r} \times {}^{0}\hat{\Delta}_{\sim i} + {}^{0}\hat{\omega}_{\sim i}^{r} \times \left[{}^{0}\hat{\omega}_{\sim i}^{r} \times {}^{0}\hat{\Delta}_{\sim i} \right] \right.$$

$$\left. \left. + 2\, {}^{0}\hat{\omega}_{\sim i}^{r} \times {}^{0}\overset{\circ}{\hat{\Delta}}_{\sim i} + {}^{0}\overset{\circ\circ}{\hat{\Delta}}_{\sim i} \right] \right.$$

$$+ \quad \frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial \dot{q}_j}\right] \cdot \left[I^{f}_{1}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + {}^{0}\hat{\underset{\sim}{\Delta}}_{1} \times {}^{0}\hat{\underset{\sim}{v}}^{r}_{1} + {}^{0}\overset{\circ}{\underset{\sim}{h}}_{1} + {}^{0}\overset{\circ}{\underset{\sim}{f}}_{1} \right]$$

$$+ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial \dot{q}_j}\right] \cdot \left[I^{f}_{1}\, {}^{0}\dot{\hat{\underset{\sim}{\omega}}}^{r}_{1} + \dot{I}^{f}_{1}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + \left({}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{1} + {}^{0}\dot{\hat{\underset{\sim}{\Delta}}}_{1} \right) \times {}^{0}\hat{\underset{\sim}{v}}^{r}_{1} \right.$$

$$\left. + {}^{0}\hat{\underset{\sim}{\Delta}}_{1} \times {}^{0}\dot{\hat{\underset{\sim}{v}}}^{r}_{1} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times \left[{}^{0}\overset{\circ}{\underset{\sim}{h}}_{1} + {}^{0}\overset{\circ}{\underset{\sim}{f}}_{1} \right] + {}^{0}\ddot{\underset{\sim}{h}}_{1} + {}^{0}\ddot{\underset{\sim}{f}}_{1} \right] \Bigg\} \qquad (2\text{-}25)$$

where ($\overset{\circ}{\cdot}$) and ($\ddot{\cdot}$) indicate first and second derivatives, respectively, with respect to local coordinates. Simplification requires the following identities:

Identity 2-1:

$$\frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{1}}{\partial \dot{q}_j}\right] = \frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{1}}{\partial q_j}$$

Identity 2-2:

$$\dot{I}^{f}_{1}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} = {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times I^{f}_{1}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + \overset{\circ}{I}^{f}_{1}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}$$

Identity 2-3:

$$\frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial \dot{q}_j}\right] = \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial q_j} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial \dot{q}_j}$$

where $\overset{\circ}{I}^{f}_{1}$ is the derivative of the inertia tensor with respect to frame (x'_1, y'_1, z'_1). Proof of identity (2-1) is straightforward. Proof of identity (2-2) follows a procedure similar to that followed by Silver [63]. Note however that:

$$\dot{I}_i^f = {}^{0}\hat{\underset{\sim}{\omega}}_i^r \times I_i^f - I_i^f \times {}^{0}\hat{\underset{\sim}{\omega}}_i^r + {}^{s}I_i^f$$

Proof of Identity (2-3):

Note that:

$${}^{0}\hat{\underset{\sim}{\omega}}_i^r = \sum_{j=1}^{i} \underset{\sim}{z}_{j-1}\,\dot{q}_j + \sum_{j=1}^{i-1}\sum_{k=1}^{m_j} \left[\delta_{jk}\right] {}^{0}\underset{\sim}{\phi}_{jk}$$

Then:

$$\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r}{\partial\, \dot{q}_j} = \underset{\sim}{z}_{j-1}, \qquad \text{for} \quad i \geq j$$

Also:

$$\frac{\partial\, \underset{\sim}{z}_{j-1}}{\partial\, t} = {}^{0}\hat{\underset{\sim}{\omega}}_{j-1} \times \underset{\sim}{z}_{j-1},$$

Hence:

$$\frac{\partial\, {}^{0}\underset{\sim}{\omega}_i^r}{\partial\, q_j} = \sum_{l=1}^{i} \frac{\partial}{\partial\, \dot{q}_j}\left[\frac{\partial\, \underset{\sim}{z}_{l-1}}{\partial\, t}\right]\dot{q}_l + \sum_{l=1}^{i-1} \frac{\partial}{\partial\, \dot{q}_j}\left[\frac{\partial\, {}^{0}\underset{\sim}{\phi}_l}{\partial\, t}\right]$$

$$= \underset{\sim}{z}_{j-1} \times \left[\sum_{l=j}^{i} \underset{\sim}{z}_{l-1}\,\dot{q}_l + \sum_{l=j}^{i-1} {}^{0}\underset{\sim}{\phi}_l\right]$$

The proof can now be written:

$$\frac{d}{dt}\left[\frac{\partial\, {}^{0}\underset{\sim}{\omega}_i^r}{\partial\, q_j}\right]$$

$$= {}^{0}\underset{\sim}{\omega}_{j-1} \times \underset{\sim}{z}_{j-1}$$

$$= \left[{}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} - \sum_{l=j}^{i} \underset{\sim}{z}_{l-1}\,\dot{q}_l - \sum_{l=j}^{i-1} {}^{0}\dot{\underset{\sim}{\phi}}_{l} \right] \times \underset{\sim}{z}_{j-1}$$

$$= {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times \underset{\sim}{z}_{j-1} + \underset{\sim}{z}_{j-1} \times \left[\sum_{l=j}^{i} \underset{\sim}{z}_{l-1}\,\dot{q}_l - \sum_{l=}^{i-} {}^{0}\dot{\underset{\sim}{\phi}}_{l} \right]$$

$$= {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times \frac{\partial\, {}^{0}\underset{\sim}{\omega}^{r}_{i}}{\partial\, \dot{q}_j} + \frac{\partial\, {}^{0}\underset{\sim}{\omega}^{r}_{i}}{\partial\, q_j} \qquad \underline{\text{End of proof.}}$$

Equation (2-25) now becomes:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{q}_j}\right] = \sum_{i}^{\cdot} \left\{ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{i}}{\partial \dot{q}_j}\right] \cdot \left[M_i\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{i} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times \left[{}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} \right] + 2\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times {}^{0}\hat{\dot{\underset{\sim}{\Delta}}}_{i} + {}^{0}\hat{\ddot{\underset{\sim}{\Delta}}}_{i} \right] \right.$$

$$+ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i}}{\partial \dot{q}_j}\right] \cdot \left[I_i^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} + \dot{I}_i^{r}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} + \left({}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} + {}^{0}\hat{\dot{\underset{\sim}{\Delta}}}_{i} \right) \times {}^{0}\hat{\underset{\sim}{v}}^{r}_{i} + {}^{0}\hat{\underset{\sim}{\Delta}}_{i} \times {}^{0}\hat{\dot{\underset{\sim}{v}}}^{r}_{i} - {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times \left({}^{0}\hat{\underset{\sim}{\Delta}}_{i} + {}^{0}\hat{\dot{\underset{\sim}{\Delta}}}_{i} \right) + {}^{0}\ddot{\underset{\sim}{h}}_{i} + {}^{0}\ddot{\underset{\sim}{f}}_{i} \right]$$

$$+ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{i}}{\partial q_j}\right] \cdot \left[M_i\, {}^{0}\hat{\underset{\sim}{v}}^{r}_{i} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} + {}^{0}\hat{\dot{\underset{\sim}{\Delta}}}_{i} \right]$$

$$\left. + \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i}}{\partial q_j}\right] \cdot \left[I_i^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{i} + {}^{0}\hat{\underset{\sim}{\Delta}}_{i} \times {}^{0}\hat{\underset{\sim}{v}}^{r}_{i} + {}^{0}\ddot{\underset{\sim}{h}}_{i} + {}^{0}\ddot{\underset{\sim}{f}}_{i} \right] \right\} \qquad (2\text{-}26)$$

The final term of (2-22) is:

$$\frac{\partial K}{\partial q_j} = \sum_i \left\{ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}_i^r}{\partial q_j} \right] \cdot \left[M_i\, {}^{0}\hat{\underset{\sim}{v}}_i^r + {}^{0}\hat{\underset{\sim}{\omega}}_i^r \times {}^{0}\hat{\underset{\sim}{\Delta}}_i + {}^{0}\hat{\underset{\sim}{\Delta}}^{\circ}_i \right] \right.$$

$$+ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r}{\partial q_j} \right] \cdot \left[I_i^f\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r + {}^{0}\hat{\underset{\sim}{\Delta}}_i \times {}^{0}\hat{\underset{\sim}{v}}_i^r + {}^{0}\dot{\underset{\sim}{h}}^{\circ}_i + {}^{0}\hat{\underset{\sim}{f}}^{\circ}_i \right]$$

$$+ \frac{1}{2}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r \cdot \frac{\partial I_i^f}{\partial q_j}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r + {}^{0}\hat{\underset{\sim}{v}}_i^r \cdot \left[\hat{\underset{\sim}{\omega}}_i^r \times \frac{\partial \hat{\underset{\sim}{\Delta}}_i}{\partial q_j} + \frac{\partial \hat{\underset{\sim}{\Delta}}^{\circ}_i}{\partial q_j} \right]$$

$$\left. + \hat{\underset{\sim}{\omega}}_i^r \times \left[\frac{\partial \hat{\underset{\sim}{h}}^{\circ}_i}{\partial q_j} + \frac{\partial \hat{\underset{\sim}{f}}^{\circ}_i}{\partial q_j} \right] \right\} \qquad (2\text{-}27)$$

Further simplifications can now be made using the following identities:

Identity 2-4:

$$\frac{1}{2}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r \cdot \frac{\partial I_i^f}{\partial q_j}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r = - \left[\hat{\underset{\sim}{\omega}}_i^r \times I_i^f\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r \right] \cdot \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r}{\partial \dot{q}_j}$$

Identity 2-5:

$$\frac{\partial \hat{\underset{\sim}{\Delta}}_i}{\partial q_j} = \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r}{\partial \dot{q}_j} \times \hat{\underset{\sim}{\Delta}}_i$$

Identity 2-6:

$$\frac{\partial \hat{\underset{\sim}{\Delta}}^{\circ}_i}{\partial q_j} = \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^r}{\partial \dot{q}_j} \times \hat{\underset{\sim}{\Delta}}^{\circ}_i$$

Identity 2-7:

$$\frac{\partial \overset{\circ}{h}_i}{\partial q_j} = \frac{\partial \, {}^{0}\hat{\omega}_i^r}{\partial \dot{q}_j} \times {}^{0}\overset{\circ}{h}_i$$

Identity 2-8:

$$\frac{\partial \overset{\circ}{f}_i}{\partial q_j} = \frac{\partial \, {}^{0}\hat{\omega}_i^r}{\partial \dot{q}_j} \times {}^{0}\overset{\circ}{f}_i$$

Identity 2-9:

$$\left[{}^{0}\hat{\omega}_i^r \times {}^{0}\hat{\Delta}_i \right] \times {}^{0}\hat{v}_i^r - {}^{0}\hat{\omega}_i^r \times \left[{}^{0}\hat{\Delta}_i \times {}^{0}\hat{v}_i^r \right] = \left[{}^{0}\hat{\omega}_i^r \times {}^{0}\hat{v}_i^r \right] \times {}^{0}\hat{\Delta}_i$$

For a proof of identity 2-4, see reference [63] and note that:

$$\frac{\partial I_i^f}{\partial \dot{q}_j} = [0].$$

Identities (2-5) to (2-8) make use of the fact that ${}^{0}\hat{\Delta}_i$, ${}^{0}\hat{\overset{\circ}{\Delta}}_i$, ${}^{0}\overset{\circ}{h}_i$, and ${}^{0}\overset{\circ}{f}_i$ are all independent of $\dot{q}_j$, so that:

$$\frac{\partial \, {}^{0}\hat{\Delta}_i}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \left[\frac{d \, {}^{0}\hat{\Delta}_i}{dt} \right]$$

and similarly for ${}^{0}\hat{\overset{\circ}{\Delta}}_i$, ${}^{0}\overset{\circ}{h}_i$, and ${}^{0}\overset{\circ}{f}_i$. Identity (2-9) is proven using the laws of triple cross products.

Noting also that $\frac{\partial \, {}^{0}\hat{\omega}_i^r}{\partial \dot{q}_j}$ and $\frac{\partial \, {}^{0}\hat{v}_i^r}{\partial \dot{q}_j}$ are zero for $j > i$,

the Lagrangian equation for joint j can finally be written as follows:

$$\underset{\sim}{\tau}_j = \sum_{i=j}^{N} \left\{ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{V}}_i^{r}}{\partial \dot{q}_j} \right] \cdot \left[M_i\, {}^{0}\hat{\underset{\sim}{V}}_i^{r} + {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} \times {}^{0}\hat{\underset{\sim}{\Delta}}_i + \right.\right.$$

$$\left. {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} \times \left({}^{0}\hat{\underset{\sim}{\omega}}_i^{r} \times {}^{0}\hat{\underset{\sim}{\Delta}}_i \right) + 2\, {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} \times {}^{0}\hat{\underset{\sim}{\delta}}_i + {}^{0}\hat{\underset{\sim}{\Delta}}_i \right]$$

$$\left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_i^{r}}{\partial \dot{q}_j} \right] \cdot \left[I_i^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} + {}^{0}\hat{I}_i^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} + {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} \times I_i^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}_i^{r} + \right.$$

$$\left. {}^{0}\hat{\underset{\sim}{\Delta}}_i \times {}^{0}\hat{\underset{\sim}{V}}_i^{r} + \hat{\underset{\sim}{\omega}}_i^{r} \times \left({}^{0}\hat{\underset{\sim}{h}}_i + {}^{0}\hat{\underset{\sim}{f}}_i \right) + {}^{0}\hat{\underset{\sim}{h}}_i + {}^{0}\hat{\underset{\sim}{f}}_i \right] \right\}$$

(2-28)

2.4.3 Flexibility Equation.

In the assumed-modes method, the generalized flexibility variables are the modal coefficients $\underset{\sim}{\delta}_{ijx}$, $\underset{\sim}{\delta}_{ijy}$ and $\underset{\sim}{\delta}_{ijz}$. The procedure for deriving the flexibility equations is similar for all three directions in which vibrations occur, and will therefore be detailed for the y-direction only. The final equations for all three directions will be given at the end.

The first two terms of the Lagrangian equation of motion for the elastic deformation of the j'th link, (equation 2-23), can be expanded in much the same way as (2-22) was expanded above. It should be noted that the third term of the kinetic energy equation

(2-21) is now dependent on the $\left[\dot{\delta}_{1k}\right]$'s, but not on the $\left[\delta_{1k}\right]$'s. The following identities apply:

Identity 2-10:

$$\frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{v}_{1}^{r}}{\partial\dot{\delta}_{jfy}}\right] = \frac{\partial\, {}^{0}\hat{v}_{1}^{r}}{\partial\delta_{jfy}}$$

Identity 2-11:

$$\frac{\partial\hat{\Delta}_{1}}{\partial\delta_{jfy}} = \frac{\partial\, {}^{0}\hat{\omega}_{1}^{r}}{\partial\dot{\delta}_{jfy}} \times \hat{\Delta}_{1}$$

Identity 2-12:

$$\frac{\partial\hat{\mathring{\Delta}}_{1}}{\partial\delta_{jfy}} = \frac{\partial\, {}^{0}\hat{\omega}_{1}^{r}}{\partial\dot{\delta}_{jfy}} \times \hat{\mathring{\Delta}}_{1}$$

Identity 2-13:

$$\frac{\partial\mathring{h}_{1}}{\partial\delta_{jfy}} = \frac{\partial\, {}^{0}\hat{\omega}_{1}^{r}}{\partial\dot{\delta}_{jfy}} \times \mathring{h}_{1}$$

Identity 2-14:

$$\frac{\partial\mathring{f}_{1}}{\partial\delta_{jfy}} = \frac{\partial\, {}^{0}\hat{\omega}_{1}^{r}}{\partial\dot{\delta}_{jfy}} \times \mathring{f}_{1}$$

Identity 2-15:

$$\frac{1}{2}\, {}^{0}\hat{\omega}_{1}^{r} \cdot \frac{\partial I_{1}^{f}}{\partial\delta_{jfy}}\, {}^{0}\hat{\omega}_{1}^{r} = -\left[{}^{0}\hat{\omega}_{1}^{r} \times I_{1}^{f}\, {}^{0}\hat{\omega}_{1}^{r}\right] \cdot \frac{\partial\, {}^{0}\hat{\omega}_{1}^{r}}{\partial\dot{\delta}_{jfy}}$$

$$+ \frac{1}{2}\, {}^{0}\hat{\omega}_{1}^{r} \cdot G_{jfy}\, {}^{0}\hat{\omega}_{1}^{r}$$

where $G_{jfy} = \dfrac{\partial^* I_j^f}{\partial \delta_{jfy}}$, the derivative being performed in local coordinates.

Identity 2-16:

$$\frac{d}{dt}\left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \dot{\delta}_{ijy}}\right] = \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \delta_{ijy}} + {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \dot{\delta}_{ijy}}$$

Identity (2-10) is similar to identity (2-1) and its proof is straightforward. Proof of identity (2-11) makes use of the fact that ${}^{0}\hat{\underset{\sim}{A}}_{i}$ is independent of $\dot{\delta}_{jfy}$, so that:

$$\frac{\partial\, {}^{0}\hat{\underset{\sim}{A}}_{i}}{\partial \delta_{ijy}} = \frac{\partial}{\partial \dot{\delta}_{ijy}}\left[\frac{d\, {}^{0}\hat{\underset{\sim}{A}}_{i}}{dt}\right]$$

To prove identity (2-12), we note that:

$${}^{0}\hat{\underset{\sim}{\dot{A}}}_{i} = \sum_{j=1}^{m_i}\left[\dot{\delta}_{ij}\right]\, {}^{0}\underset{\sim}{e}_{ij}$$

and that ${}^{0}\underset{\sim}{e}_{ij}$ is independent of $\left[\dot{\delta}_{ij}\right]$. Hence:

$$\frac{\partial\, {}^{0}\hat{\underset{\sim}{A}}_{i}}{\partial \delta_{jfy}} = \sum_{k=1}^{m_j}\left[\dot{\delta}_{jk}\right]\, \frac{d\, {}^{0}\underset{\sim}{e}_{ij}}{dt}$$

$$= \frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \dot{\delta}_{jfy}} \times \hat{\underset{\sim}{A}}_{i}$$

Identities (2-13) to (2-15) can be proved in a similar manner and will therefore not be presented.

Proof of Identity 2-16:

Note first that:

$$\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \dot{\delta}_{jfy}} = \begin{bmatrix} 0 \\ \phi_{jfy} \\ 0 \end{bmatrix} \triangleq \underset{\sim}{\Phi}_{jfy} \; ; \qquad i > j$$

and:

$$\frac{\partial\, {}^{0}\underset{\sim}{\omega}_{i}^{r}}{\partial \delta_{jfy}} = \sum_{l=1}^{i} \frac{\partial}{\partial \dot{\delta}_{jfy}} \left[\frac{\partial \underset{\sim}{z}_{l-1}}{\partial t} \right] \dot{q}_{l}$$

$$+ \sum_{l=1}^{i-1} \sum_{l=1}^{i-1} \left[\dot{\delta}_{jk} \right] \frac{\partial}{\partial \dot{\delta}_{jfy}} \left[\frac{\partial \underset{\sim}{\phi}_{lk}}{\partial t} \right]$$

$$= \underset{\sim}{\Phi}_{jfy} \times \left[\sum_{l=j+1}^{i} \underset{\sim}{z}_{l-1} \dot{q}_{l} + \sum_{l=j+1}^{i-1} \dot{\underset{\sim}{\phi}}_{l} \right]$$

for $i > j$.

The proof can now be written:

$$\frac{d}{dt} \left[\frac{\partial\, {}^{0}\underset{\sim}{\omega}_{i}^{r}}{\partial \dot{\delta}_{ijf}} \right]$$

$$= {}^{0}\underset{\sim}{\omega}_{j-1} \times \underset{\sim}{\Phi}_{jf},$$

$$= \left[{}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} - \sum_{i=j+1}^{i} \underset{\sim}{z}_{i-1}\,\dot{q}_{i} - \sum_{i=j+1}^{i-1} \overset{\circ}{\underset{\sim}{\phi}}_{i} \right] \times \underset{\sim}{\Phi}_{jfy}$$

$$= {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times \underset{\sim}{\Phi}_{jfy} + \underset{\sim}{\Phi}_{jfy} \times \left[\sum_{i=j+1}^{i} \underset{\sim}{z}_{i-1}\,\dot{q}_{i} - \sum_{i=j+1}^{i-1} \overset{\circ}{\underset{\sim}{\phi}}_{i} \right]$$

$$= {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times \frac{\partial\, {}^{0}\underset{\sim}{\omega}_{i}^{r}}{\partial \overset{\circ}{\delta}_{ijy}} + \frac{\partial\, {}^{0}\underset{\sim}{\omega}_{i}^{r}}{\partial \dot{\delta}_{ijy}}$$

End of Proof.

Following a procedure similar to that used for the joint equation, and also making use of Identity (2-2), the first two terms of (2-23) can be expanded and the result simplified to yield:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{\delta}_{ijy}}\right] - \frac{\partial K}{\partial \delta_{ijy}} = \sum_{i=j}^{N} \Bigg\{ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{v}}_{i}^{r}}{\partial \dot{\delta}_{ijy}}\right] \cdot \left[M_{i}\, {}^{0}\ddot{\underset{\sim}{v}}_{i}^{r} + {}^{0}\dot{\hat{\underset{\sim}{\omega}}}_{i}^{r} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} + {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times \left({}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{i} \right) + 2\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times {}^{0}\overset{\circ}{\hat{\underset{\sim}{\Delta}}}_{i} + {}^{0}\ddot{\hat{\underset{\sim}{\Delta}}}_{i} \right]$$

$$+ \left[\frac{\partial\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r}}{\partial \dot{\delta}_{ijy}}\right] \cdot \left[I_{i}^{f}\, {}^{0}\dot{\hat{\underset{\sim}{\omega}}}_{i}^{r} + \overset{\circ}{I}_{i}^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} + {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} \times I_{i}^{f}\, {}^{0}\hat{\underset{\sim}{\omega}}_{i}^{r} + {}^{0}\hat{\underset{\sim}{\Delta}}_{i} \times {}^{0}\ddot{\underset{\sim}{v}}_{i}^{r} + \dot{\hat{\underset{\sim}{\omega}}}_{i}^{r} \times \overset{\circ}{\underset{\sim}{h}}_{i} + \overset{\circ}{\underset{\sim}{f}}_{i} + \ddot{\underset{\sim}{h}}_{i} + \ddot{\underset{\sim}{f}}_{i} \right] \Bigg\}$$

$$+ t'_{jfy} \qquad (2\text{-}29)$$

where:

$$t'_{jfy} = \hat{\underset{\sim}{V}}^{r}_{i} \cdot \frac{\partial \dot{\underset{\sim}{\Delta}}^{\circ}_{i}}{\partial \dot{\delta}_{ijy}} + \hat{\underset{\sim}{\omega}}^{r}_{i} \cdot \left[\frac{\partial \dot{\underset{\sim}{h}}^{\circ}_{i}}{\partial \dot{\delta}_{ijy}} + \frac{\partial \dot{\underset{\sim}{f}}^{s}_{i}}{\partial \dot{\delta}_{ijy}} \right]$$

$$+ \hat{\underset{\sim}{\omega}}^{r}_{i} \cdot \left[\frac{d}{dt} \left[\frac{\partial \underset{\sim}{f}^{s}_{i}}{\partial \dot{\delta}_{ijy}} \right] + \frac{\partial \underset{\sim}{f}^{s}_{i}}{\partial \delta_{ijy}} \right] + \frac{d}{dt} \left[\frac{\partial \underset{\sim}{b}^{\circ}_{i}}{\partial \dot{\delta}_{ijy}} \right]$$

$$- \frac{1}{2} \hat{\underset{\sim}{\omega}}^{r}_{i} \cdot \frac{\partial \underset{\sim}{I}^{f}_{i}}{\partial \delta_{ijy}} \hat{\underset{\sim}{\omega}}^{r}_{i} \qquad (2\text{-}30)$$

The terms that constitute t'_{jfy} involve derivatives in local coordinates only and are very easy to simplify. For example,

$$\frac{\partial \dot{\underset{\sim}{\Delta}}^{\circ}_{i}}{\partial \dot{\delta}_{ijy}} = \begin{bmatrix} 0 \\ \bar{\Delta}_{jfy} \\ 0 \end{bmatrix}$$

Expressions for t'_{jfx} and t'_{jfz} can be derived in a similar manner.

2.4.4 Elastic Potential Energy Term.

In this section, we examine the last (third) term of equation (2.23). Elastic potential energy is accounted for, to a good approximation, by bending about the transverse y and z directions and by torsion about the longitudinal x-axis. For the i^{th} link, the potential energy is given by the expression:

$$P_{e_i} = \frac{1}{2}\int_{\text{link } i} \left\{ EI_z \left[\frac{\partial^2 \Delta_{zi}}{\partial x^2} \right]^2 + EI_y \left[\frac{\partial^2 \Delta_{yi}}{\partial x^2} \right]^2 + GI_x \left[\frac{\partial^2 \Phi_{xi}}{\partial x} \right]^2 \right\} dx$$

where:

Δ_{zi}, and Δ_{yi} are deflections in the indicated directions,

Φ_{xi} is the rotation about the x-axis due to torsion,

E is Young's modulus of elasticity of the material,

G is the shear modulus of elasticity of the material,

I_x is the polar area moment of elasticity,

I_y , I_z is the area moment of inertia of the link cross-section, about the y and z axes respectively.

When the truncated modal series are inserted for the deflections and rotations in the above equation, the last term of (2-23) can be written in the following form:

$$\frac{\partial P_e}{\partial \delta_{jfy}} = \sum \left[\delta_{jk} \right] K_{jkfy} \qquad (2\text{-}31)$$

where K_{jkfy} is a scalar stiffness coefficient which can be calculated from the elastic constants of the links and the mode-shapes for flexure and torsion. Details of this procedure are given by Book [4] and are not unique to the development presented here. Similar expressions hold for the potential energy term of the flexibility equation in the x and z directions.

Equations (2-30) and (2-31) can be combined as follows:

$$t_{jfy} = \hat{\underset{\sim}{V}}^{r}_{1} \cdot \frac{\partial \dot{\underset{\sim}{\Delta}}_{1}}{\partial \dot{\delta}_{ijy}} + \hat{\underset{\sim}{\omega}}^{r}_{1} \cdot \left[\frac{\partial \overset{\circ}{\underset{\sim}{h}}_{1}}{\partial \dot{\delta}_{ijy}} + \frac{\partial \overset{s}{\underset{\sim}{f}}_{1}}{\partial \dot{\delta}_{ijy}} \right]$$

$$+ \hat{\underset{\sim}{\omega}}^{r}_{1} \cdot \left[\frac{d}{dt} \left[\frac{\partial \overset{s}{\underset{\sim}{f}}_{1}}{\partial \dot{\delta}_{ijy}} \right] + \frac{\partial \overset{s}{\underset{\sim}{f}}_{1}}{\partial \delta_{ijy}} \right] + \frac{d}{dt} \left[\frac{\partial \overset{\circ}{b}_{1}}{\partial \dot{\delta}_{ijy}} \right]$$

$$- \frac{1}{2} \hat{\underset{\sim}{\omega}}^{r}_{1} \cdot \frac{\partial I^{f}_{1}}{\partial \delta_{ijy}} \hat{\underset{\sim}{\omega}}^{r}_{1} + \sum \left[\delta_{jk} \right] K_{jkfy} \qquad (2\text{-}32)$$

The complete flexibility equation can therefore be written as follows:

$$\tau_{jfy} = \sum_{i=j}^{N} \left\{ \left[\frac{\partial \, {}^{0}\hat{\underset{\sim}{V}}^{r}_{1}}{\partial \dot{\delta}_{ijy}} \right] \cdot \left[M_{1} \, {}^{0}\hat{\underset{\sim}{V}}^{r}_{1} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{1} + \right. \right.$$

$$\left. {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times \left({}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times {}^{0}\hat{\underset{\sim}{\Delta}}_{1} \right) + 2 \, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times {}^{0}\dot{\underset{\sim}{\Delta}}_{1} + {}^{0}\ddot{\underset{\sim}{\Delta}}_{1} \right]$$

$$+ \left[\frac{\partial \, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1}}{\partial \dot{\delta}_{ijy}} \right] \cdot \left[I^{r}_{1} \, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + I^{f}_{1} \, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} \times I^{r}_{1} \, {}^{0}\hat{\underset{\sim}{\omega}}^{r}_{1} + \right.$$

$$\left. \left. {}^{0}\hat{\underset{\sim}{\Delta}}_{1} \times {}^{0}\hat{\underset{\sim}{V}}^{r}_{1} + \hat{\underset{\sim}{\omega}}^{r}_{1} \times \left({}^{0}\overset{\circ}{\underset{\sim}{h}}_{1} + {}^{0}\overset{s}{\underset{\sim}{f}}_{1} \right) + {}^{0}\overset{\circ}{\underset{\sim}{h}}_{1} + {}^{0}\overset{s}{\underset{\sim}{f}}_{1} \right] \right\}$$

$$+ t_{jfy} \qquad (2\text{-}33)$$

2.5. Computational Algorithm for the Inverse Dynamics.

An algorithm similar to the recursive Newton-Euler algorithm for rigid manipulators results from the joint equation (2-28). The flexibility equation (2-33) departs only slightly from this form. Both equations can be combined into a single, compact, computational procedure in which all terms, including flexibility terms, are calculated. To see how the computational algorithm is derived, note that:

$$\frac{\partial {}^{0\hat{}}\underset{\sim}{r}_i}{\partial \dot{q}_j} = \underset{\sim}{z}_{j-1}, \qquad i \geq j \tag{2-34}$$

and:

$$\frac{\partial {}^{0\hat{}}\dot{\underset{\sim}{r}}_i}{\partial \dot{q}_j} = \underset{\sim}{z}_{j-1} \times {}^{0\hat{}}\underset{\sim}{r}_{j-1,i}, \quad \text{for } i \geq j \tag{2-35}$$

where ${}^{0\hat{}}\underset{\sim}{r}_{j-1,i}$ is the vector from the origin of coordinate frame $(x_{j-1}, y_{j-1}, z_{j-1})$ to the centre of mass of the i'th link in its undeformed state. That is:

$${}^{0\hat{}}\underset{\sim}{r}_{j-1,i} = \sum_{k=j}^{i} {}^{0}\underset{\sim}{r}_k^{*} + \sum_{k=j}^{i-1} {}^{0}\underset{\sim}{\Delta}_k + {}^{0\hat{}}\underset{\sim}{s}_i \tag{2-36}$$

Note also that:

$$\frac{\partial {}^{0\hat{}}\underset{\sim}{\omega}_i}{\partial \delta_{jfy}} = \begin{bmatrix} 0 \\ 0 \\ \Phi_{jfy} \end{bmatrix} \triangleq {}^{0}\underset{\sim}{\Phi}_{jfy}; \quad \text{for} \quad i > j$$

$$\frac{\partial {}^{0}\hat{v}_{i}^{r}}{\partial \dot{\delta}_{jfy}} = \begin{bmatrix} 0 \\ 0 \\ \Phi_{jfy} \end{bmatrix} \times \hat{p}_{j-1,i}^{r} + \begin{bmatrix} 0 \\ \bar{\Delta}_{jfy} \\ 0 \end{bmatrix}$$

$$\triangleq {}^{0}\Phi_{jfy} \times {}^{0}\hat{p}_{j-1,i}^{r} + {}^{0}\Delta_{jfy} \quad \text{for } i > j;$$

The recursive equations for calculating the joint torques from the end-effector backwards to the base are therefore:

$${}^{0}f_{i} = {}^{0}f_{i+1} + {}^{0}F_{i} \tag{2-37}$$

$${}^{0}n_{i} = {}^{0}n_{i+1} + \left[{}^{0}p_{i}^{r\bullet} + {}^{0}\Delta_{i}\right] \times {}^{0}f_{i+1} + \left[{}^{0}p_{i}^{r\bullet} + {}^{0}\hat{s}_{i}^{r}\right] \times {}^{0}F_{i}$$

$$+ {}^{0}N_{i} \tag{2-38}$$

$$\tau_{j} = z_{j-1} \cdot {}^{0}n_{i} \tag{2-39}$$

where:

$${}^{0}F_{i} = \left[M_{i} {}^{0}\hat{v}_{i}^{r} + {}^{0}\hat{\omega}_{i}^{r} \times {}^{0}\hat{\Delta}_{i} + {}^{0}\hat{\omega}_{i}^{r} \times \left[{}^{0}\hat{\omega}_{i}^{r} \times {}^{0}\hat{\Delta}_{i}^{\cdot}\right] + \right.$$

$$\left. 2\, {}^{0}\hat{\omega}_{i}^{r} \times {}^{0}\hat{\dot{\Delta}}_{i} + {}^{0}\hat{\ddot{\Delta}}_{i} \right] \tag{2-40}$$

$${}^{0}N_{i} = \left[I_{i}^{f} {}^{0}\hat{\omega}_{i}^{r} + \dot{I}_{i}^{f} {}^{0}\hat{\omega}_{i}^{r} + {}^{0}\hat{\omega}_{i}^{r} \times I_{i}^{f} {}^{0}\hat{\omega}_{i}^{r} + {}^{0}\hat{\Delta}_{i} \times {}^{0}\hat{v}_{i}^{r} + \right.$$

$$\left. {}^{0}\hat{\omega}_{i}^{r} \times \left[{}^{0}h_{i}^{\circ} + {}^{0}\hat{f}_{i}^{\circ}\right] + {}^{0}\ddot{h}_{i} + {}^{0}\ddot{f}_{i} \right] \tag{2-41}$$

The flexibility equation can also be written as follows:

$$\tau_{jfy} = t_{jfy} + {}^{0}\underset{\sim}{\Phi}_{jfy} \cdot {}^{0}\underset{\sim}{n}_{j+1} + {}^{0}\underset{\sim}{\Delta}_{jfy} \cdot {}^{0}\underset{\sim}{f}_{j+1} \qquad (2\text{-}42)$$

Similar equations hold for τ_{jfx} and τ_{jfz}.

In this thesis, we consider actuator forces and torques as being applied only at the manipulator joints. In this case, τ_{jfy} is set to zero in the above equation.

Note the similarity of equations (2-35) to (2-39), to the Newton-Euler formulation of rigid manipulator equations. Equation (2-35) is identical to its rigid counterpart, and represents the equilibrium of forces for each link in translation. Equation (2-36) represents equilibrium of torques and now takes into consideration the deflection of the link's endpoint away from its normal, undeflected position.

Equation (2-38) is Newton's equation of motion of the link in translation. In addition to the inertia force due to rigid-body translation (the first term of (2-38)), the inertia force due to acceleration of the link's actual center of mass away from its undeflected position is now included (represented by the rest of the terms in (2-38)). In total, (2-38) is just the link's mass multiplied by the total velocity of the link's actual center of mass as it performs its total motion.

It is clear that equation (2-39) represents Euler's equation of rotation about the link's normal, undeflected center of mass. The first two terms are the same as the terms of the corresponding equation in the rigid-manipulator formulation. The third term accounts for the rate of change of the link's inertia as it vibrates. The fourth term accounts for the fact that the actual

center of mass is not necessarily coincident with the link's normal, undeflected center of mass. It is difficult to attribute physical meaning to the rest of the terms in this equation. It is for this reason that the Lagrangian approach has been preferred over a direct Newton-Euler approach. The computational procedure, however, possesses the same desirable attributes that are associated with Newton-Euler approaches. It is not unlikely that a direct Newton-Euler approach in which distributed link flexibility is modeled by a modal approach as in this chapter, would result in a similar, if not identical, recursive computational procedure.

Note that in (2-37) to (2-42), the vector quantities have all been referred to base coordinates. It is a simple matter to show that both the forward and backward recursive equations can be re-written as follows:

Forward expressions:

$$ {}^{1}\underset{\sim}{\omega}_{1}^{r} = {}^{1}A_{1-1}\left[{}^{1-1}\underset{\sim}{\omega}_{1-1} + \underset{\sim}{z}_0 \dot{q}_1 \right] \qquad (2\text{-}43) $$

$$ {}^{1}\underset{\sim}{\omega}_{1} = E_1^{rf}\left[{}^{1}\underset{\sim}{\omega}_{1}^{r} + {}^{1}\dot{\underset{\sim}{\phi}}_{1} \right] \qquad (2\text{-}44) $$

$$ {}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} = {}^{1}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{\omega}}_{1-1} + {}^{1-1}\underset{\sim}{\omega}_{1-1} \times \underset{\sim}{z}_0 \dot{q}_1 + \underset{\sim}{z}_0 \ddot{q}_1 \right] \qquad (2\text{-}45) $$

$$ {}^{1}\dot{\underset{\sim}{\omega}}_{1} = E_1^{rf}\left[{}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\dot{\underset{\sim}{\phi}}_{1} + {}^{1}\ddot{\underset{\sim}{\phi}}_{1} \right] \qquad (2\text{-}46) $$

$$ {}^{1}\underset{\sim}{v}_1^r = {}^{1}A_{1-1}\left[{}^{1-1}\underset{\sim}{v}_{1-1} + {}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{p}_1\right] \qquad (2\text{-}47) $$

$$ {}^{1}\underset{\sim}{v}_1 = E_1^{rf}\left[{}^{1}\underset{\sim}{v}_1^r + {}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{\Delta}_1 + {}^{1}\ddot{\underset{\sim}{\Delta}}_1\right] \qquad (2\text{-}48) $$

$$ {}^{1}\dot{\underset{\sim}{v}}_1^r = {}^{1}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{v}}_{1-1}\right] + {}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{p}_1^{r\bullet} + {}^{1}\underset{\sim}{\omega}_1^r \times ({}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{p}_1^{r\bullet}) \qquad (2\text{-}49) $$

$$ {}^{1}\dot{\underset{\sim}{v}}_1 = E_1^{rf}\left[{}^{1}\dot{\underset{\sim}{v}}_1^r + {}^{1}\dot{\underset{\sim}{\omega}}_1^r \times {}^{1}\underset{\sim}{\Delta}_1 + {}^{1}\underset{\sim}{\omega}_1^r \times ({}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{\Delta}_1) + 2({}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{\Delta}_1) + {}^{1}\ddot{\underset{\sim}{\Delta}}_1\right] \qquad (2\text{-}50) $$

$$ {}^{1}\hat{\underset{\sim}{\omega}}_{1j}^r = {}^{1}\underset{\sim}{\omega}_1^r \qquad (2\text{-}51) $$

$$ {}^{1}\hat{\dot{\underset{\sim}{\omega}}}_{1j}^r = {}^{1}\dot{\underset{\sim}{\omega}}_1^r \qquad (2\text{-}52) $$

$$ {}^{1}\hat{\underset{\sim}{v}}_{1j}^r = {}^{1}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{v}}_{1-1}\right] + {}^{1}\hat{\underset{\sim}{\omega}}_1^r \times \left({}^{1}\underset{\sim}{p}_1^{r\bullet} + {}^{1}\hat{\underset{\sim}{s}}_1\right) + {}^{1}\hat{\underset{\sim}{\omega}}_{1j}^r \times \left({}^{1}\hat{\underset{\sim}{\omega}}_1^r \times \left({}^{1}\underset{\sim}{p}_1^{r\bullet} + {}^{1}\hat{\underset{\sim}{s}}_1\right)\right) \qquad (2\text{-}53) $$

$$ {}^{1}\underset{\sim}{F}_1 = \left[M_1\, {}^{1}\hat{\underset{\sim}{v}}_1^r + {}^{1}\hat{\underset{\sim}{\omega}}_1^r \times {}^{1}\hat{\underset{\sim}{\Delta}}_1 + {}^{1}\hat{\underset{\sim}{\omega}}_1^r \times \left({}^{1}\hat{\underset{\sim}{\omega}}_1^r \times {}^{1}\hat{\underset{\sim}{\Delta}}_1\right) + 2\,{}^{1}\hat{\underset{\sim}{\omega}}_1^r \times {}^{1}\hat{\dot{\underset{\sim}{\Delta}}}_1 + {}^{1}\hat{\ddot{\underset{\sim}{\Delta}}}_1\right] \qquad (2\text{-}54) $$

$$ {}^{1}\underset{\sim}{N}_1 = \left[{}^{1}I_1^f\, {}^{1}\hat{\dot{\underset{\sim}{\omega}}}_1^r + {}^{1}I_1^f\, {}^{1}\hat{\underset{\sim}{\omega}}_1^r + {}^{1}\hat{\underset{\sim}{\omega}}_1^r \times {}^{1}I_1^f\, {}^{1}\hat{\underset{\sim}{\omega}}_1^r + {}^{1}\hat{\underset{\sim}{\Delta}}_1 \times {}^{1}\hat{\underset{\sim}{v}}_1^r \right.
$$

$$+ \; {}^{i}\hat{\underset{\sim}{\omega}}^{r}_{i} \times \left[{}^{i}\underset{\sim}{h}^{s}_{i} + {}^{i}\underset{\sim}{f}^{s}_{i} \right] + {}^{i}\underset{\sim}{h}^{m}_{i} + {}^{i}\underset{\sim}{f}^{m}_{i} \Big] \qquad (2\text{-}55)$$

The expressions for t_{jfx}, t_{jfy} and t_{jfz} are also calculated on the forward recursion.

The backward expressions are:

$$ {}^{i}\underset{\sim}{f}_{i} = E_{i} \; {}^{i}A_{i+1} \; {}^{i+1}\underset{\sim}{f}_{i+1} + {}^{i}\underset{\sim}{F}_{i} \qquad (2\text{-}56)$$

$$ {}^{i}\underset{\sim}{n}_{i} = E_{i} \; {}^{i}A_{i+1} \; {}^{i+1}\underset{\sim}{n}_{i+1} + \left[{}^{i}\underset{\sim}{p}^{r*}_{i} + {}^{i}\underset{\sim}{\Delta}_{i} \right] \times E_{i} \; {}^{i}A_{i+1} \; {}^{i+1}\underset{\sim}{f}_{i+1}$$

$$+ \left[{}^{i}\underset{\sim}{p}^{r*}_{i} + {}^{i}\hat{\underset{\sim}{s}}^{r}_{i} \right] \times {}^{i}\underset{\sim}{F}_{i} + {}^{i}\underset{\sim}{N}_{i} \qquad (2\text{-}57)$$

$$ \tau_{j} = \underset{\sim}{z}_{0} \cdot {}^{j-1}A_{j} \; \underset{\sim}{n}_{j} \qquad (2\text{-}58)$$

The flexibility equation can also be written as follows:

$$ \tau_{jfx} = t_{jfx} + {}^{j}\underset{\sim}{\Phi}_{jfx} \cdot E_{i} \; {}^{j}A_{j+1} \; {}^{j+1}\underset{\sim}{n}_{j+1} \qquad (2\text{-}59)$$

$$ \tau_{jfy} = t_{jfy} + {}^{j}\underset{\sim}{\Phi}_{jfy} \cdot E_{i} \; {}^{j}A_{j+1} \; {}^{j+1}\underset{\sim}{n}_{j+1} +$$

$$+ \; {}^{j}\underset{\sim}{\Delta}_{jfy} \cdot E_{i} \; {}^{j}A_{j+1} \; {}^{j+1}\underset{\sim}{f}_{j+1} \qquad (2\text{-}60)$$

$$ \tau_{jfz} = t_{jfz} + {}^{j}\underset{\sim}{\Phi}_{jfz} \cdot E_{i} \; {}^{j}A_{j+1} \; {}^{j+1}\underset{\sim}{n}_{j+1} +$$

$$+ \; {}^{j}\underset{\sim}{\Delta}_{jfz} \cdot E_{i} \; {}^{j}A_{j+1} \; {}^{j+1}\underset{\sim}{f}_{j+1} \qquad (2\text{-}61)$$

To include gravity effects, simply set ${}^{0}\underset{\sim}{\dot{v}}_0$ equal to the 3-dimensional gravity vector appropriate for the coordinate system of the manipulator base.

Note that $(E_i^{rf})^{-1}$ is equal to $(E_i^{rf})^T$ since E_i^{rf} is a matrix of direction cosines. Inversion of the matrix is therefore unnecessary.

2.6. Computational Algorithm for Dynamic Simulation.

In this section we derive a fast computational algorithm for dynamic simulation of flexible manipulators which could be very useful not only in manipulator design but also in control.

The manipulator dynamic equations can be assembled in the following form:

$$H \ddot{\underset{\sim}{z}} = \underset{\sim}{R} \qquad (2\text{-}62)$$

where:

$\underset{\sim}{z}$ = vector of generalized coordinates,

$= [q_1 \;\; \delta_{11} \; \delta_{12} \cdots \delta_{1m_1} \;\; q_2 \;\; \delta_{21} \; \delta_{22} \cdots \delta_{2m_2} \cdots \;\; \delta_{Nm_N}]$

H = Inertia matrix in the order for multiplication appropriate for $\underset{\sim}{z}$,

$\underset{\sim}{R}$ = vector of remaining dynamics and external forcing terms.

When $\underset{\sim}{z}$ is arranged in the order shown here, the resulting inertia matrix H is symmetric. Numerical simulation of the motion of a flexible manipulator entails solution of (2-62) to obtain the generalized accelerations $\ddot{\underset{\sim}{z}}$, which are then integrated twice to yield the generalized velocities and coordinates. The problem now

is to find H and R. To do this, we follow the method of Walker[44].

$\underset{\sim}{R}$ is simply the difference between the external forcing vector and the dynamics obtained from the inverse plant algorithm when all the generalized accelerations set to zero.

To calculate H, note that if the inverse plant algorithm is run with all references to gravity, velocities, elastic forces and external loads eliminated, and with $\underset{\sim}{z} = e_k$, where e_k is a vector the same length as $\underset{\sim}{z}$, with all elements equal to zero except a '1' in the k'^{th} location, then the resulting dynamics from the joint and flexibility equations would form a vector that would be equal to the k'^{th} column of H. This can be seen from (2-62). Hence all columns of H could be found by successively incrementing the value of k. This, however, is an inefficient procedure. An alternative is to follow a procedure similar to that of Walker's method 3.

To describe the forward dynamics algorithm, we shall use the following notation for terms of the H matrix:

$h_{i:j}$ = the term in the column corresponding to $\ddot{q}_j$ and the row corresponding to $\ddot{q}_i$;

$h_{i:\delta ik?}$ = the term in the column corresponding to $\ddot{\delta}_{ik?}$ and the row corresponding to $\ddot{q}_i$

$h_{\delta ik?:j}$ = the term in the column corresponding to $\ddot{q}_j$ and the row corresponding to $\ddot{\delta}_{ik?}$;

$h_{\delta ik?:\delta ij?}$ = the term in the column corresponding to $\ddot{\delta}_{ij?}$ and the row corresponding to $\ddot{\delta}_{ik?}$;

where ? refers to either x, y or z. In the algorithm that follows, we assume that the manipulator payload is a concentrated mass. Then, starting with:

M_{N+1} = payload mass.

${}^{N}\underset{\sim}{c}_{N+1} = \underset{\sim}{0}$

${}^{N}\underset{\sim}{g}_{N+1} = [0]$

calculate the following for the j^{th} link during the recursion from the last (N^{th}) link down to the manipulator base:

$$M_j = M_{j+1} + m_j, \tag{2-63}$$

$$M_j\,{}^{j}\underset{\sim}{c}_j = M_{j+1}\left[{}^{j}\underset{\sim}{p}_j^{r^*} + {}^{j}\underset{\sim}{\Delta}_j + {}^{j}\underset{\sim}{c}_{j+1}\right] + m_j\left[{}^{j}\underset{\sim}{p}_j^{r^*} + {}^{j}\hat{\underset{\sim}{s}}_j\right] + {}^{j}\hat{\underset{\sim}{\Delta}}_j \tag{2-64}$$

$${}^{(j-1)}\underset{\sim}{c}_j = E_{j-1}^{fr}\;{}^{(j-1)}A_j\;{}^{j}\underset{\sim}{c}_j\;{}^{j}A_{j-1}\;E_{j-1}^{rf} \tag{2-65}$$

$$\underset{\sim}{X}_1 = {}^{j}\underset{\sim}{c}_{j+1} + {}^{j}\underset{\sim}{p}_j^{r^*} + {}^{j}\underset{\sim}{c}_j \tag{2-66}$$

$$\underset{\sim}{X}_2 = {}^{j}\underset{\sim}{p}_j^{r^*} + {}^{j}\hat{\underset{\sim}{s}}_j + \frac{{}^{j}\hat{\underset{\sim}{\Delta}}_j}{m_j} - {}^{j}\underset{\sim}{c}_j \tag{2-67}$$

$${}^{(j-1)}\underset{\sim}{g}_j = E_{j-1}^{fr}\;{}^{j-1}A_j\left\{{}^{j}\underset{\sim}{g}_{j+1} + M_{j+1}\,F(\underset{\sim}{X}_1) + {}^{j}I_j - F({}^{j}\hat{\underset{\sim}{\Delta}}_j)/m_j + m_j\,F(\underset{\sim}{X}_2)\right\}{}^{j}A_{j-1}E_{j-1}^{rf} \tag{2-68}$$

where $F(\underset{\sim}{X}) = (\underset{\sim}{X}\cdot\underset{\sim}{X})I - \underset{\sim}{X}\,\underset{\sim}{X}^{T}$.

For the k^{th} flexibility mode, form the following vectors:

A column corresponding to the generalized variable δ_{jky} is calculated by forming the following vectors:

$$ {}^{j}f_{j+1:\delta jkx} = M_{j+1}\left[{}^{j}\underset{\sim}{\Phi}_{jkx} \times {}^{j}c_{j+1} \right] \qquad (2\text{-}69) $$

$$ {}^{j}f_{j+1:\delta jky} = M_{j+1}\left[{}^{j}\underset{\sim}{\Delta}_{jky} + {}^{j}\underset{\sim}{\Phi}_{jky} \times {}^{j}c_{j+1} \right] \qquad (2\text{-}70) $$

$$ {}^{j}f_{j+1:\delta jkz} = M_{j+1}\left[{}^{j}\underset{\sim}{\Delta}_{jkz} + {}^{j}\underset{\sim}{\Phi}_{jkz} \times {}^{j}c_{j+1} \right] \qquad (2\text{-}71) $$

where:

$$ {}^{j}\underset{\sim}{\Delta}_{jky} = \begin{bmatrix} 0 \\ {}^{j}\Delta_{jky} \\ 0 \end{bmatrix} \qquad (2\text{-}72) $$

$$ {}^{j}\underset{\sim}{\Delta}_{jkz} = \begin{bmatrix} 0 \\ 0 \\ {}^{j}\Delta_{jky} \end{bmatrix} \qquad (2\text{-}73) $$

$$ {}^{j}\underset{\sim}{\Phi}_{jkx} = \begin{bmatrix} {}^{j}\phi_{jkx} \\ 0 \\ 0 \end{bmatrix} \qquad (2\text{-}74) $$

$$ {}^{j}\underset{\sim}{\Phi}_{jky} = \begin{bmatrix} 0 \\ 0 \\ {}^{j}\phi_{jkx} \end{bmatrix} \qquad (2\text{-}75) $$

$$ {}^{j}\underset{\sim}{\Phi}_{jkz} = \begin{bmatrix} 0 \\ {}^{j}\phi_{jkx} \\ 0 \end{bmatrix} \qquad (2\text{-}76) $$

Also, calculate the following vectors:

$$ {}^{j}n_{j+1:\delta jkx} = {}^{j}g_{j+1}\,{}^{j}\Phi_{jkx} + {}^{j}c_{j+1} \times {}^{j}f_{j+1:\delta jkx} \qquad (2\text{-}77) $$

$$ {}^{j}n_{j+1:\delta jky} = {}^{j}g_{j+1}\,{}^{j}\Phi_{jky} + {}^{j}c_{j+1} \times {}^{j}f_{j+1:\delta jky} \qquad (2\text{-}78) $$

$$ {}^{j}n_{j+1:\delta jkz} = {}^{j}g_{j+1}\,{}^{j}\Phi_{jkz} + {}^{j}c_{j+1} \times {}^{j}f_{j+1:\delta jkz} \qquad (2\text{-}79) $$

To form a column corresponding to the variable δ_{jkx}, the diagonal term is given by the assignment:

$$ h_{jkx:jkx} = b_{jkx} + {}^{j}\Phi_{jkz} \cdot {}^{j}n_{j+1:\delta jkx} \qquad (2\text{-}80) $$

The terms in this column above the diagonal element are obtained by following a recursive procedure from this mode backwards through the preceeding modes of the same link, then through the preceeding links, all the way down to the manipulator base. The recursive equations are the equivalents of equations (2-53) to (2-58).

A similar procedure is followed in order to calculate the terms of the columns corresponding to the variables δ_{jky} and δ_{jkz}.

The terms of a column corresponding to the joint variable q_j are obtained by first forming the following vectors:

$$ {}^{j-1}f_j = E_j^{rf}\, z_0 \times M_j\, {}^{j-1}c_j \qquad (2\text{-}81) $$

$$ {}^{j-1}n_j = {}^{j-1}g_j\, E_j^{rf}\, z_0 + {}^{j-1}c_j \times {}^{j-1}f_j \qquad (2\text{-}82) $$

where $z_0 = [0 \;\; 0 \;\; 1]^T$. The diagonal term of the column is given by:

$$h_{j:j} = \text{z-component of } \left[E_j^{rf\ j-1}\underset{\sim}{n}_j\right] \qquad (2\text{-}83)$$

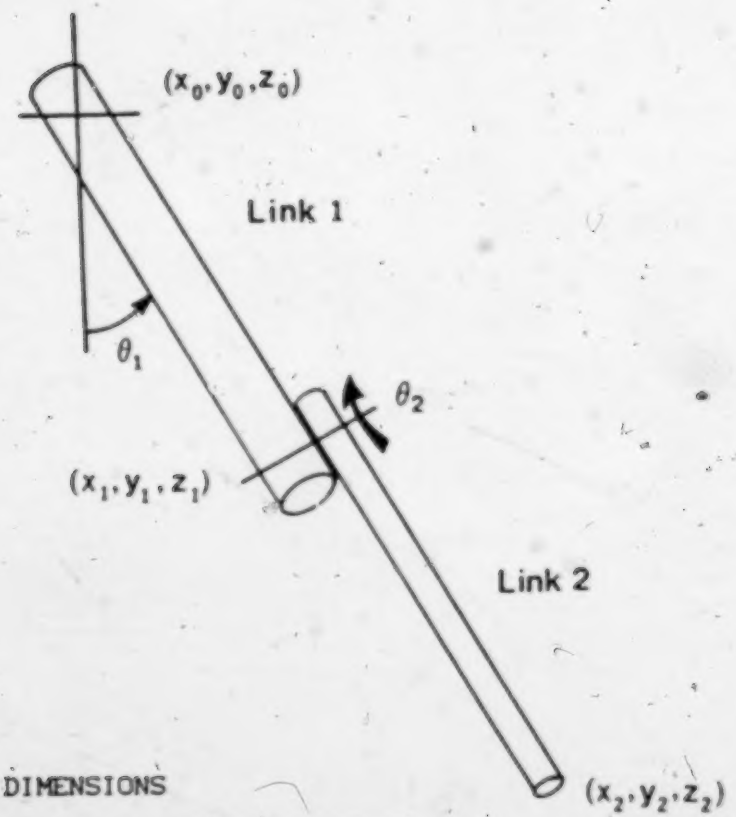
The elements in this column that are above the diagonal element are calculated by a backward recursion from the j^{th} joint down to the base of the manipulator. The recursive equations are, as for the flexibility variable case, obtained by modifying equations (2-53) to (2-58) for this case.

This completes the algorithm for calculating the inertia matrix.

2.7. Simulation Example.

The forward dynamic algorithm is demonstrated by simulation of the two-link spatial flexible manipulator shown in Figure 2-2. The manipulator dimensions are also given in Figure 2.2. The initial values of the deflection variables are all set to zero, and the initial joint angles are: q_1 = -80 degrees, and q_2 = 10 degrees. The arm is allowed to swing freely under gravity. The results are shown in Figure 2.3, which shows the oscillations of the joint angles ("JOINT 1" and "JOINT 2"), as well as the vibrations of the end-point of each link in the y ("YDISP"), z ("ZDISP") and x ("TORSION") directions. Note that there is no torsion of the second link.

An important observation that can be made from the simulation results is that the oscillations that arise as a result of structural flexibility appear superimposed on top of lower-frequency, gross joint-angle oscillations. This can be seen in the plots in Figure 2.3 of "YDISP" and "ZDISP" for both links.



DIMENSIONS

	Link 1	Link 2
Length	1.22 m	1.22 m
Radius	0.005 m	0.004 m
Material	Aluminum	Aluminum
Density	2.78×10^3 kg-m^3	2.78×10^3 kg-m^3
Young's Modulus	6.9×10^{10} N-m^{-2}	6.9×10^{10} N-m^{-2}
Shear Modulus	8.9×10^9 N-m^{-2}	8.9×10^9 N-m^{-2}
EI product	33.9 N-m^2	13.9 N-m^2
GJ product	78.5 N-m^2	32.9 N-m^2
Frequency	5.9 hz	4.7 hz

Figure 2.2. Example Manipulator Used in the Simulations.

This observation is the basis for the control algorithm that is proposed in Chapter 3. If link-flexibility is not too great, then two well separated sets of time constants are present in the flexible manipulator system. Model decomposition into slow and fast subsystems is therefore possible. The control system design problem becomes more tractable in that designs can now be based upon the two simpler subsystems individually, and the results combined into a composite control system. This is the approach taken in Chapter 4 where composite controllers are proposed and tested by simulation.

2.8. Comparison with Book's Recursive Lagrangian Method.

Of all the published papers on algorithms for numerical modeling of flexible manipulators, Book's paper on recursive Lagrangian dynamics [4] has been the only one to include an analysis of computational complexity. In order to determine the relative speed and efficiency of the algorithms presented in this chapter, we perform an analysis of their computational complexity and compare the results with Book's.

Book's algorithms are mainly applicable to planar manipulators. This restriction results from his use of the same generalized variables to represent elastic deflections in all three planes. This can hardly be correct, since motion in one plane does not necessarily imply motion in another. Comparison with his method necessarily requires a reformulation of the fully general, 3-D algorithms presented here to include the assumptions made by Book. This is quite easily done by replacing every occurrence of the diagonal matrix "$\mathrm{diag}[\delta_{xik}, \delta_{yik}, \delta_{zik}]$" by the

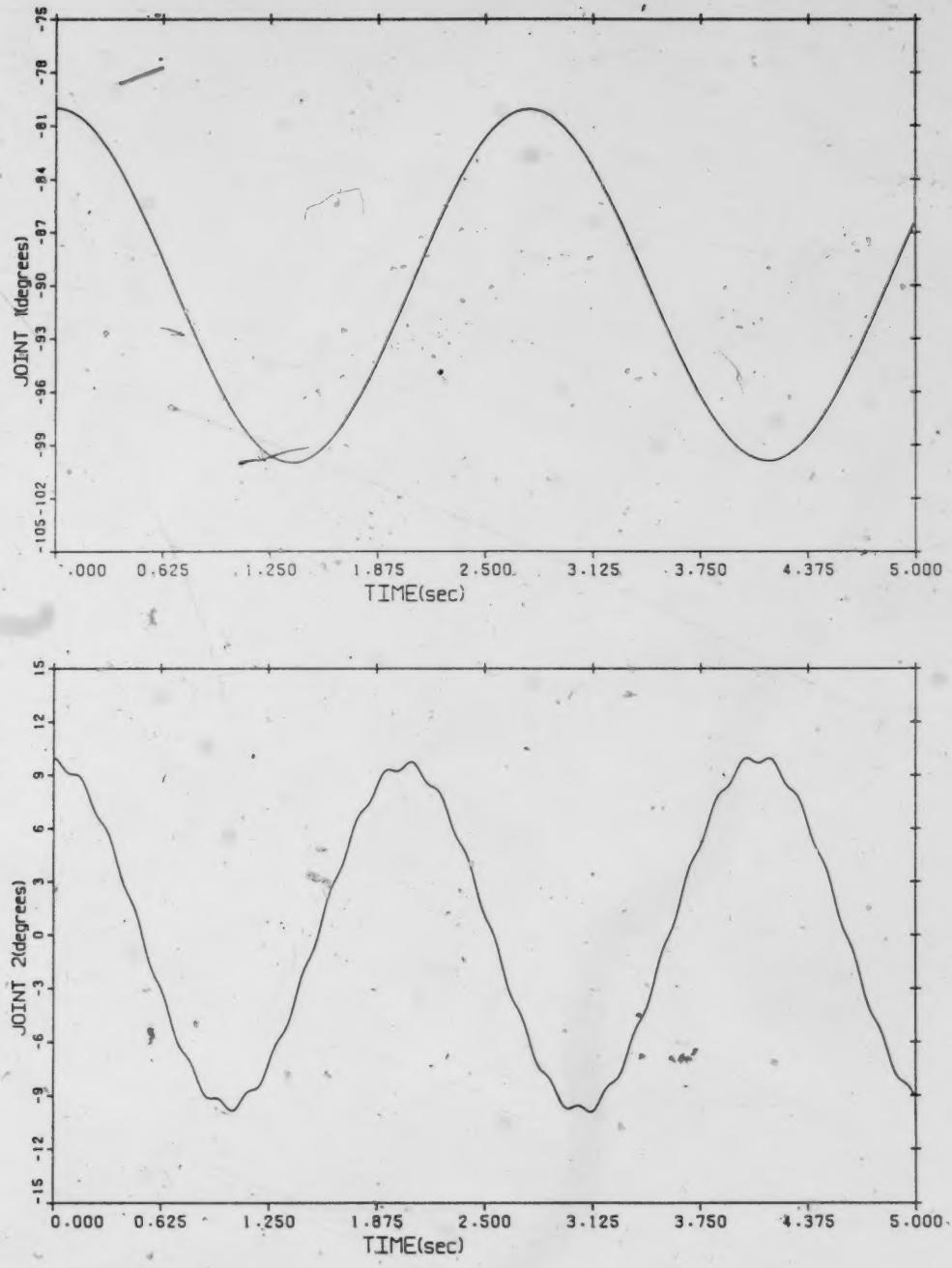
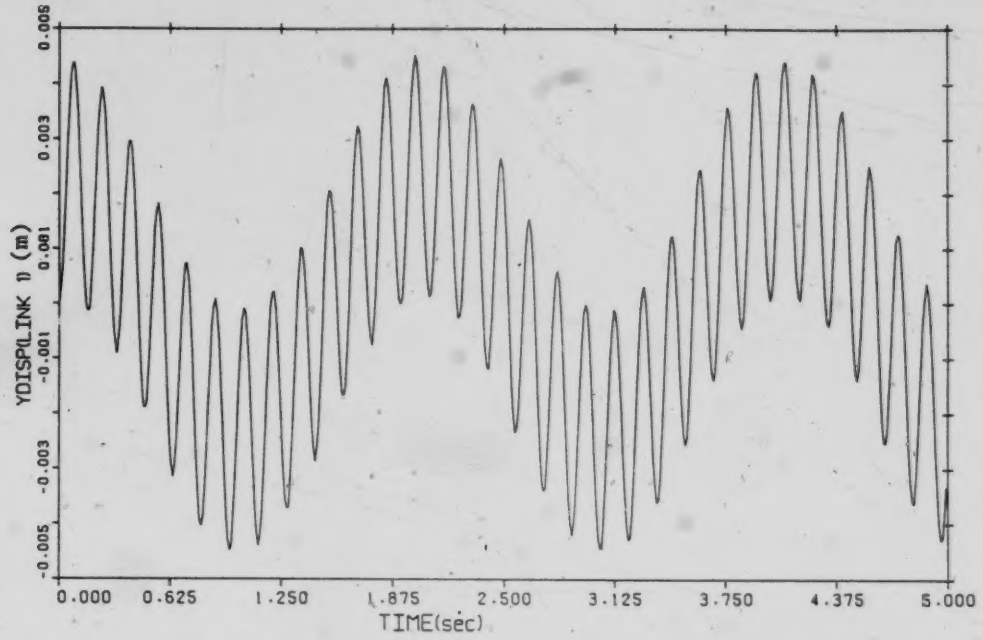


Figure 2.3. Simulation Results of the Two-Link Manipulator. (Labels are explained in the text).



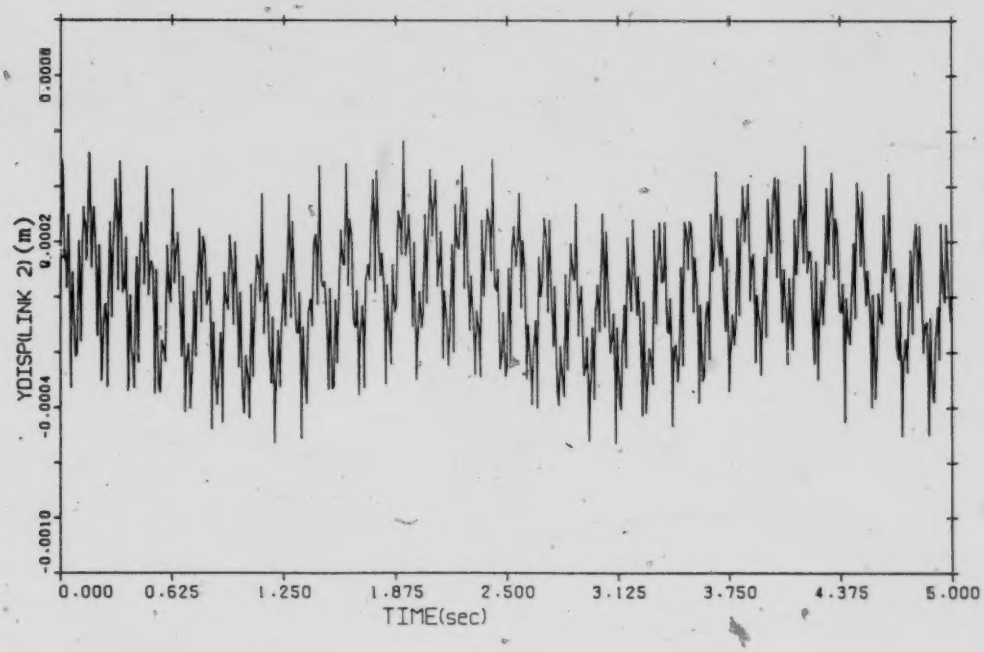
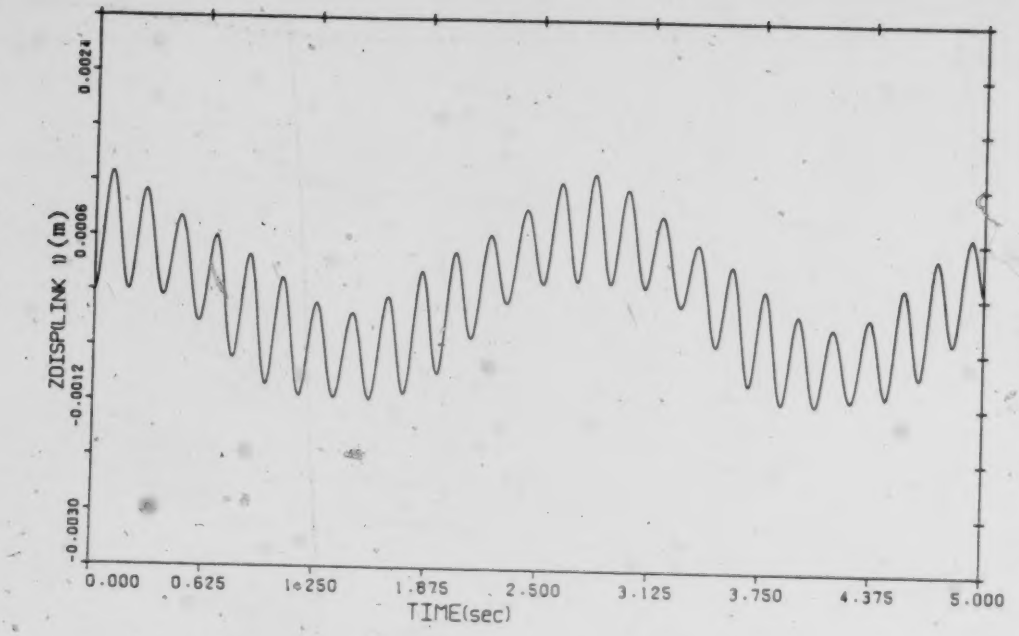


Figure 2.3. (Cont'd)



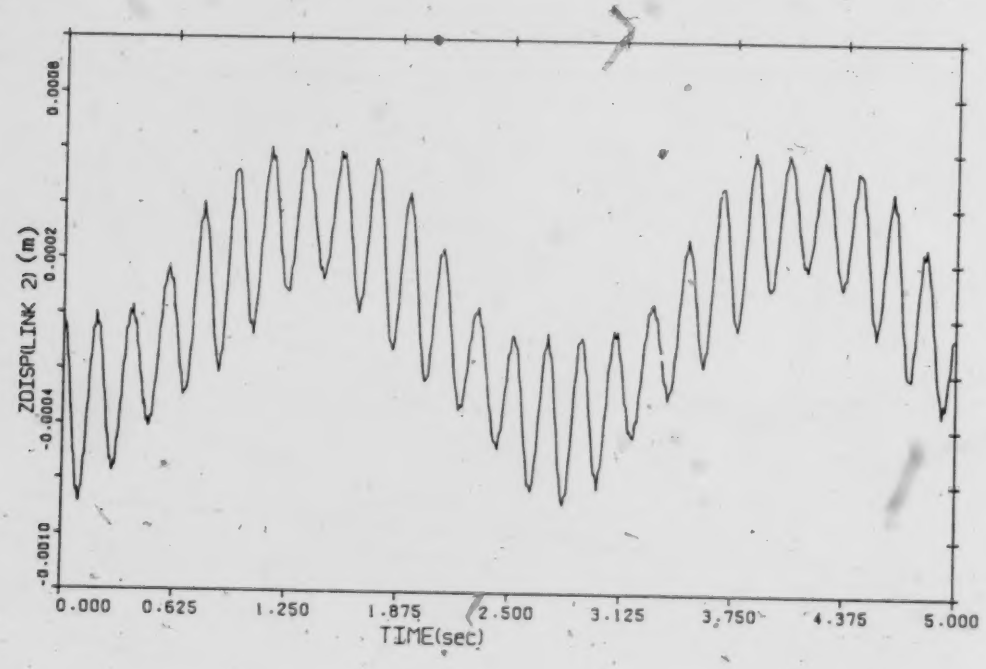


Figure 2.3. (Cont'd)

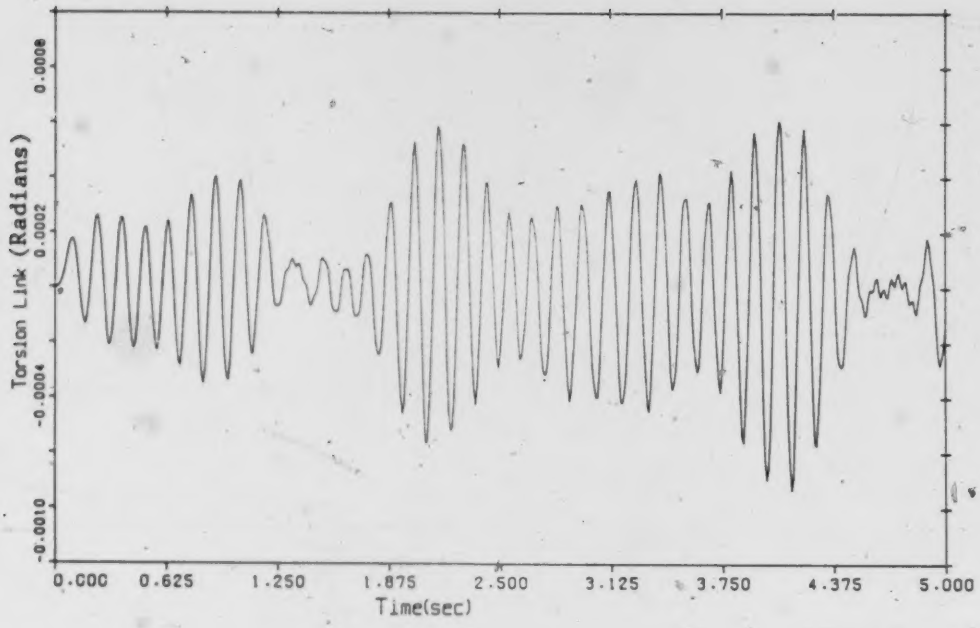


Figure 2.3. (Cont'd)

scalar "δ_{ik}". Significant simplification of the algorithms result. Comparison with Book's method is now more meaningful. Numerical results obtained from simulation of an example arm moving in a plane, using the resulting algorithmic equations, are identical to those obtained using Book's method and to those obtained using the general 3-D equations.

In order to compare the speeds of execution of the two dynamic formulations, we compare the total number of multiplications and the total number of additions required to compute H and R in equation (2-59). Book gives a pair of expressions for these numbers that are derived after the following assumptions were made: (1) Obvious simplifications in matrix multiplications due to the special structure of matrices or vectors in these computations were made, (2) terms that appeared in multiple equations were calculated only once and saved, (3) the integrations required to compute I_f^f, etc., were not included. When we make the same assumptions, the numbers of multiplications and additions required for one complete run of the inverse plant algorithm are as follows:

$$\text{No. of Mults.} = 32nm^2 + 105nm + 300n - 93$$

$$\text{No. of Adds.} = 35nm^2 + 49nm + 244n - 91$$

The numbers of computations required to calculate H are:

$$\text{No. of Mults.} = 3n^2m^2 + nm^2 + 16.5n^2m + 13.5n^2 - 2.5m^2 + 43.5nm + 118.5n - 65.5m - 95$$

$$\text{No. of Adds.} = 3n^2m^2 + 1.5nm^2 + 13.5n^2m + 10n^2 - 3m^2$$

$$+ 42.5nm + 108n - 66m - 86$$

where:

n = number of links, all flexible,

m = number of modes per link, same for all links.

Note that the dependency on m^2 in the inverse plant algorithm is due to the double summations over the modes of vibration that are required. For a manipulator with 6 links, all of which are flexible, and with 3 modes per link, the scheme proposed in this chapter requires a total of 9,799 multiplications and 8,064 additions to compute H and R in equation (2-59). Book's method requires 34,593 multiplications, or about 3.5 times as many as the scheme of this chapter and 33,653 additions, or 4.1 times as many as the scheme of this chapter. This clearly demonstrates the success of the method in terms of speed and accuracy.

If Walker's method 3 were applied to a 6-link flexible manipulator, and we model the three modes of vibration of each link by giving each link an additional 3 degrees of freedom, then the total number of multiplications required to compute H and R of (2-62) would be 11,495. This number is greater than the number obtained when the algorithm presented in this chapter is used to model the same manipulator with three modes per link. Furthermore, the method of this chapter yields more accurate numerical results.

Book states that in order to be competitive with possible Newton-Euler, non-transfer matrix approaches, he would have to make certain simplifications of the assumed mode shapes. The same simplifications can be made for the mode shapes here as well. This

will result in further speeding up the algorithms presented in this chapter.

The algorithms presented here revert to Luh's Newton-Euler algorithm for links that are considered rigid. Unnecessary equations should not be computed for such links. For example, if the i'th link is rigid then ${}^{i}\dot{v}_i = {}^{i}\dot{v}_i^r$, hence it is unnecessary to compute equation (2-50), for example, for this link. When the algorithm is programmed in this way, the number of multiplications required to compute H and R of equation (2-59) for a 6-link manipulator with two flexible links and three modes per flexible link is 3,190, compared with 11,361 for Book's method.

2.9. Discussion and Summary.

The algorithms for modeling flexible manipulators that have been presented in this chapter are successful in terms of speed of computation. How closely the numerical model represents the actual manipulator depends, to a large extent, on the accuracy to which the flexible deflections are modeled. The assumption that the infinite modal expansion can be truncated has been discussed by several authors [87,88]. The general conclusion has been made that in most cases, good control action can still be achieved if the controller is based on a model in which the first few mode shapes are represented fairly accurately. Alberts et al [5] have shown that the higher vibratory modes can also be damped by wrapping the links with a visco-elastic material. When this is done, the accuracy of truncated models becomes quite good.

Most authors make the assumption that link oscillations are relatively small since it allows for consideration of linear,

elastic beam vibrations only. Cetinkunt [89] suggested that one way to ensure this in practice is to develop trajectory generation algorithms that design tasks such that excitation of flexible states minimized.

Advanced schemes for on-line control of flexible manipulators would undoubtedly require real-time calculation of some, if not all, of the terms of the inverse and forward dynamic equations. The efficiency of the computational procedure presented in this chapter makes it attractive for on-line implementation of, say, some form of modified, computed-torque control scheme for flexible manipulators that might be based on reduced-order models. In order to calculate the recursive inverse dynamics algorithm that was presented in this chapter for a six-link robot arm with two flexible links and three modes per link, current 16-bit microprocessors require 5-10 milliseconds. Even this is still too long for real-time control of flexible manipulators. Availability of faster microprocessors in the near future should make real-time control feasible.

Another advantage of the inverse dynamics formulation of this chapter is that it can capitalize on the methods for parallel computation of the rigid Newton-Euler algorithm that have appeared in the literature [55-62]. It seems that these methods can be extended quite easily to include this algorithm, hence considerably speeding up the calculations.

In summary, a Newton-Euler-like algorithm is presented in this chapter for numerical calculation of inverse and forward dynamics of flexible manipulators. Inverse dynamics are calculated by first calculating kinematics on a forward recursion from the manipulator

base to the tip then calculating dynamics on the return recursion. The forward dynamics formulation involves a direct procedure for calculating the full inertia matrix including flexibility effects. The algorithm for calculating forward dynamics is at least three to four times as fast as the only other method for which computation times are available in the literature, to date, and just as accurate. A discussion of some practical implementational problems in flexible manipulator control is also presented. Despite these obstacles, the methods presented in this chapter should prove very useful in designing the mechanical and control systems of flexible manipulators.

CHAPTER 3. RECURSIVE FINITE ELEMENT MODELING OF FLEXIBLE MANIPULATORS

3.1. Introduction.

The two approaches to flexible manipulator modeling that have been identified as the most promising to date are the Assumed-Modes method[4], and the Finite-element method[6]. In the assumed-modes method, which is discussed in Chapter 2, link-deflections are represented by the superposition of products of deflection mode shapes and time-dependent generalized coordinates. Simple boundary conditions are assumed at the joints and these are used to derive analytic expressions for the mode shapes. In reality, the boundary conditions may be more complicated and there has been some uncertainty as to whether the simple, clamped-clamped or clamped-free mode shapes can satisfactorily represent flexible-link kinematics. In the finite-element method, which is the subject of this chapter, this problem does not exist. The actual boundary conditions can be more accurately represented in the finite-element equations.

This advantage of the finite-element method, however, is not gained without a price. This method is noted for requiring long CPU times for problem solution. Two factors contribute to these high computational requirements. Firstly, the Lagrangian approach that is used in most finite-element programs to calculate dynamic terms leads to numerical algorithms that are inefficient [41]. Secondly, the finite-element method itself results in a very high

model dimension. For example, four modes of vibration may be sufficient to describe link deflections in a flexible manipulator using the assumed-modes method, but the finite element method may require ten or more elements, resulting in fifty or more degrees of freedom per link.

Fast and efficient algorithms are needed in order to quickly simulate alternative manipulator mechanical designs for the flexible manipulator. Even greater speed is required for real-time implementation of the control system since feedforward terms may have to be calculated in order to achieve decoupling of flexibility effects from rigid-body dynamics. In this chapter we present new, recursive algorithms that alleviate the problem of computational inefficiency of the traditional Lagrangian Finite-element formulation of flexible manipulator dynamics. These algorithms are derived in a manner similar to the manner in which the recursive, assumed modes algorithms are derived in Chapter 2. Velocities and accelerations are calculated on a forward recursion from the manipulator base to its tip, and joint torques and flexibility dynamic terms are calculated on the return recursion. The traditional finite-element representation of flexible-link kinematics is retained. The method produces the same numerical results as the Lagrangian finite element method but the algorithm is much faster and more efficient.

In the sections that follow, the development of the algorithms is presented in some detail. The numerical results obtained from simulation of the same two-link, spatial flexible manipulator described in chapter 2 are presented. A method of estimating mode shapes from finite element data is then given and a comparison is

made with the fundamental mode shape assuming clamped-free boundary conditions.

3.2. Finite-Element Kinematics.

In the approach taken in this chapter, the i'th link is divided into n_i linear elements, with the j'th element having length ℓ_{ij} (see Figure 3.1). For simplicity we will assume that each element is of constant cross-sectional dimensions, with Young's moduli E_{ijy} and E_{ijz} in the y and z direction respectively, and shear modulus G_{ijx} about the longitudinal x-axis. An irregularly shaped element can be divided into smaller elements to conform with this assumption. Each element has assigned to it a local, "rigid" coordinate system $(x'_{ij}, y'_{ij}, z'_{ij})$, with origin located at the element's distal end when it is undeformed, as shown in Figure 3.1. Note that x'_{ij} lies along the undeformed length of the element. Displacements and rotations of the element are assumed to be measured with respect to this coordinate frame. Associated with each element are five time-dependent generalized coordinates, namely, the element-tip's deflections in the y (u_{ijy})and z (u_{ijz}) directions, their derivatives ($\varphi_{ijy}$and φ_{ijz}) with respect to x'_{ij}, and the element-tip's angular rotation (φ_{ijx}) about the x'_{ij} axis due to torsion.

A notable characteristic of the formulation presented here is the assumption that link-oscillations are small in magnitude compared to the link's length. This assumption is not necessary in order to derive recursive algorithms but it is a realistic representation of the situation that would most likely exist in

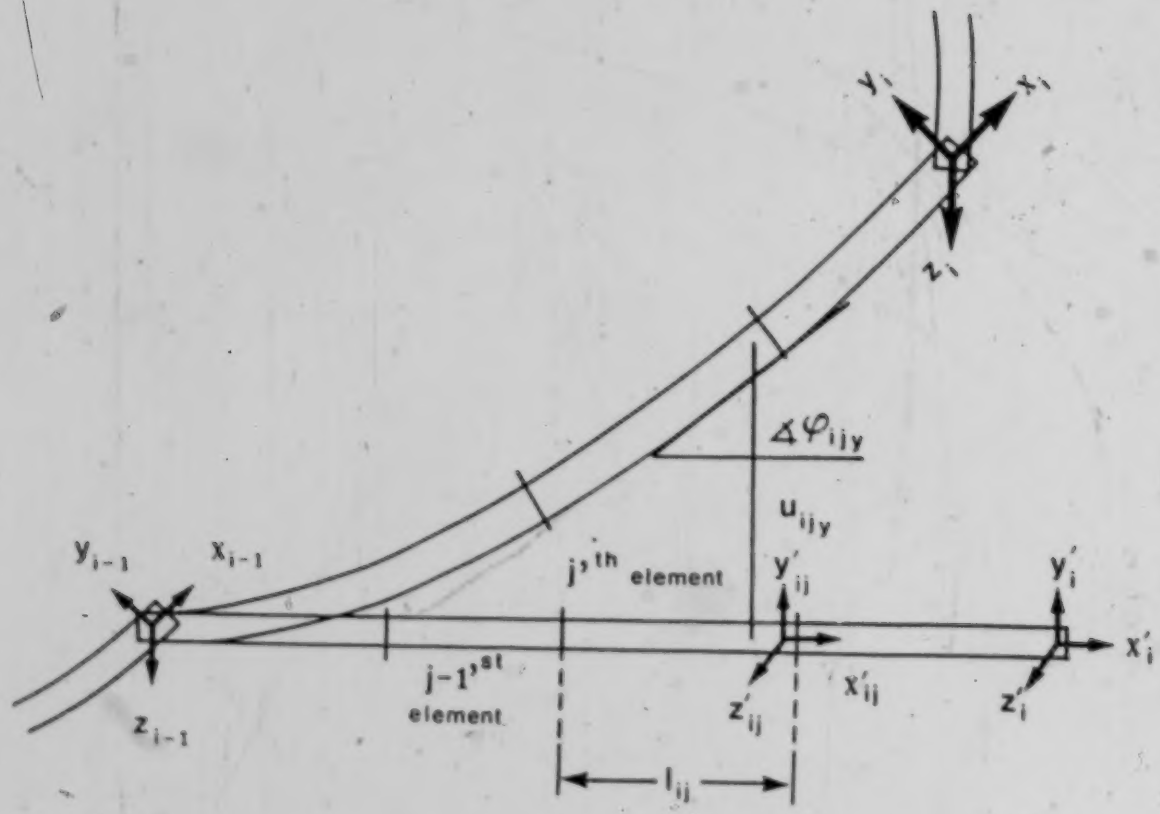


Figure 3.1. Element Definitions for Finite Element Model of the i^{th} link.

the majority of practical manipulators of the near future. Terms that contain higher order powers of deflection variables can therefore be neglected. The resulting simplification of dynamic terms is significant. Inaccuracies introduced by this assumption are relatively small [6] and, in a control system design, can be compensated for by feedback. The coupling effects between flexural deflections and torsional angles are neglected, since these are at least of second order. Higher-order terms involving only flexural deflections are retained in the current formulation but in an actual simulation or controller design, these terms can be neglected as well.

Flexural deflections of interior points of an element, in both the y and z directions, are interpolated by third-order polynomials [38]. Thus:

$$\begin{aligned} {}^{i'}\Delta_{ijy}(x'_{ij}) &= P_1(x'_{ij})\, u_{i(j-1)y} + P_2(x'_{ij})\, \varphi_{i(j-1)y} \\ &\quad + P_3(x'_{ij})\, u_{ijy} + P_4(x'_{ij})\, \varphi_{ijy} \\ &\triangleq \underset{\sim}{P}^{T}_{ijy}(x'_{ij})\, \underset{\sim}{V}_{ijy} \end{aligned} \tag{3-1}$$

where:

$$P_1(x) = 1 - 3\frac{x^2}{\ell^2} + 2\frac{x^3}{\ell^3} \tag{3-2a}$$

$$P_2(x) = x - 2\frac{x^2}{\ell} + \frac{x^3}{\ell^2} \tag{3-2b}$$

the majority of practical manipulators of the near future. Terms that contain higher order powers of deflection variables can therefore be neglected. The resulting simplification of dynamic terms is significant. Inaccuracies introduced by this assumption are relatively small [6] and, in a control system design, can be compensated for by feedback. The coupling effects between flexural deflections and torsional angles are neglected, since these are at least of second order. Higher-order terms involving only flexural deflections are retained in the current formulation but in an actual simulation or controller design, these terms can be neglected as well.

Flexural deflections of interior points of an element, in both the y and z directions, are interpolated by third-order polynomials [38]. Thus:

$$
\begin{aligned}
{}^{i'}\Delta_{ijy}(x'_{ij}) = {} & P_1(x'_{ij})\, u_{i(j-1)y} + P_2(x'_{ij})\, \varphi_{i(j-1)y} \\
& + P_3(x'_{ij})\, u_{ijy} + P_4(x'_{ij})\, \varphi_{ijy} \\
\triangleq {} & \underset{\sim}{P}^{T}_{ijy}(x'_{ij})\, \underset{\sim}{\nu}_{ijy} \qquad (3\text{-}1)
\end{aligned}
$$

where:

$$P_1(x) = 1 - 3\frac{x^2}{\ell^2} + 2\frac{x^3}{\ell^3} \qquad (3\text{-}2a)$$

$$P_2(x) = x - 2\frac{x^2}{\ell} + \frac{x^3}{\ell^2} \qquad (3\text{-}2b)$$

$$P_3(x) = 3 - \frac{x^2}{\ell^2} - 2\frac{x^3}{\ell^3} \qquad (3\text{-}2c)$$

$$P_4(x) = -\frac{x^2}{\ell} + \frac{x^3}{\ell^3} \qquad (3\text{-}2d)$$

The left superscript (i') indicates that the parameter is measured with respect to the "primed" coordinate system $(x'_{ij}, y'_{ij}, z'_{ij})$. Similar expressions obtain for ${}^{i'}\Delta_{ijz}$ in the z-direction. Torsional rotations ${}^{i'}\phi_{ijx}(x'_{ij})$ are interpolated by linear polynomials, hence:

$${}^{i'}\phi_{ijx}(x'_{ij}) = Q_1(x'_{ij})\ \varphi_{i(j-1)x} + Q_2(x'_{ij})\ \varphi_{ijx}$$

$$\triangleq \underset{\sim}{Q}^{T}_{ijx}(x'_{ij})\ \underset{\sim}{V}_{ijx} \qquad (3\text{-}3)$$

where:

$$Q_1(x) = 1 - \frac{x}{\ell} \qquad (3\text{-}4a)$$

$$Q_2(x) = \frac{x}{\ell} \qquad (3\text{-}4b)$$

Higher order interpolation polynomials can be used to represent torsional vibrations at interior points but this may not be necessary unless torsion is significant.

It is useful to define a coordinate frame (x'_i, y'_i, z'_i) with its origin located at the link's tip when it is undeformed, as shown in Figure 3-1. The axes of this frame are located according to Paul's convention [19] exactly as described in chapter 1. The usual kinematic parameters that describe rigid-link motion are

defined, such as the link-twist, α_i, link length, ℓ_i, offset d_i, and joint rotational angle q_i. Motion of this coordinate frame represents the gross, rigid-body motion of the manipulator.

As mentioned in chapter 1, an additional set of kinematic parameters is required to describe elastic oscillations about the rigid-link motion. In order to define this new set, we need to define another coordinate frame (x_i, y_i, z_i) with its origin located at the link's actual tip and moving with the link as it executes both its gross (rigid) motion and elastic oscillations. (See Figure 3-1). The axes are oriented such that this coordinate frame becomes identical to the rigid frame (x'_i, y'_i, z'_i) when the link is undeformed. Kinematics of the link's tip due to flexibility can now be represented in terms of the angles ${}^{i'}\phi_{ix}$, ${}^{i'}\phi_{iy}$ and ${}^{i'}\phi_{iz}$, which are the angular displacements of the x, y and z axes respectively of frame (x_i, y_i, z_i) from (x'_i, y'_i, z'_i), and in terms of linear displacements ${}^{i'}\Delta_{iy}$ and ${}^{i'}\Delta_{iz}$ of the origin of (x_i, y_i, z_i) from the origin of (x'_i, y'_i, z'_i). For a link with $(x'_{i,n_i}, y'_{i,n_i}, z'_{i,n_i})$ parallel to (x'_i, y'_i, z'_i), the following hold:

$$ {}^{i'}\phi_{ix} = \varphi_{i,n_i x} $$

$$ {}^{i'}\phi_{iy} = \varphi_{i,n_i y} $$

$$ {}^{i'}\phi_{iz} = \varphi_{i,n_i z} $$

$$ {}^{i'}\Delta_{iy} = {}^{i'}\Delta_{i,n_i y} $$

$$^{i'}\Delta_{iz} = {}^{i'}\Delta_{i,n_i z}$$

The rotational transformation E_i^{fr} that transforms frame (x_i, y_i, z_i) to frame (x'_i, y'_i, z'_i) is already defined in chapter 1 as:

$$E_i^{fr} = \text{Rot}(x, {}^{i'}\phi_{ix})\ \text{Rot}(y, {}^{i'}\phi_{iy})\ \text{Rot}(z, {}^{i'}\phi_{iz}) \qquad (3\text{-}5)$$

This transformation matrix, together with the joint rotational matrix $^{i-1}A_{i'}$, allow rotational kinematics to be referred from one coordinate system to another on the same or any other link, and finally to the inertial system (x_0, y_0, z_0) located at the manipulator base. Definitions of E_i^{rf} and $^{i-1}A_{i'}$ are the same as E_i^{rf} and $^{i-1}A_i$ as shown in Figure 1-3.

Recursive expressions can now be written for angular velocities and accelerations ($^{i'}\underset{\sim}{\omega}_i^r$, $^{i'}\dot{\underset{\sim}{\omega}}_i^r$), and linear velocities and accelerations ($^{i'}\underset{\sim}{v}_i^r$, $^{i'}\dot{\underset{\sim}{v}}_i^r$) of the rigid-coordinate frame (x'_i, y'_i, z'_i) of each link. The left superscript "i'" indicates that the quantity is referred to the rigid coordinate system. Recursive expressions can also be written for the angular velocities and accelerations ($^{i}\underset{\sim}{\omega}_i$, $^{i}\dot{\underset{\sim}{\omega}}_i$), and the linear velocities and acceleration ($^{i}\underset{\sim}{v}_i$, $^{i}\dot{\underset{\sim}{v}}_i$) of frame (x_i, y_i, z_i) of each link. The left superscript "i" indicates that the quantity is referred to frame (x_i, y_i, z_i). The expressions follow:

$$^{i'}\underset{\sim}{\omega}_i^r = {}^{i'}A_{i-1}\left[{}^{i-1}\underset{\sim}{\omega}_{i-1} + \underset{\sim}{z}_0\,\dot{q}_i\right] \qquad (3\text{-}6a)$$

$$^{i}\underset{\sim}{\omega}_i = E_i^{rf}\left[{}^{i'}\underset{\sim}{\omega}_i^r + {}^{i'}\underset{\sim}{\phi}_i\right] \qquad (3\text{-}6b)$$

$$ {}^{1'}\dot{\underset{\sim}{\omega}}_1^r = {}^{1'}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{\omega}}_{1-1} + {}^{1-1}\underset{\sim}{\omega}_{1-1} \times \underset{\sim}{z}_0\dot{q}_1 + \underset{\sim}{z}_0\ddot{q}_1\right] \quad (3\text{-}6c) $$

$$ {}^{1}\dot{\underset{\sim}{\omega}}_1 = E_1^{rf}\left[{}^{1'}\dot{\underset{\sim}{\omega}}_1^r + {}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\mathring{\underset{\sim}{\phi}}_1 + {}^{1'}\ddot{\underset{\sim}{\phi}}_1\right] \quad (3\text{-}6d) $$

$$ {}^{1'}\underset{\sim}{v}_1^r = {}^{1'}A_{1-1}\left[{}^{1-1}\underset{\sim}{v}_{1-1} + {}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{p}_1\right] \quad (3\text{-}6e) $$

$$ {}^{1}\underset{\sim}{v}_1 = E_1^{rf}\left[{}^{1'}\underset{\sim}{v}_1^r + {}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{\Delta}_1 + {}^{1'}\ddot{\underset{\sim}{\Delta}}_1\right] \quad (3\text{-}6f) $$

$$ {}^{1'}\dot{\underset{\sim}{v}}_1^r = {}^{1'}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{v}}_{1-1}\right] + {}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{p}_1^{r\bullet} + {}^{1'}\underset{\sim}{\omega}_1^r \times ({}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{p}_1^{r\bullet}) \quad (3\text{-}6g) $$

$$ {}^{1}\dot{\underset{\sim}{v}}_1 = E_1^{rf}\left[{}^{1'}\dot{\underset{\sim}{v}}_1^r + {}^{1'}\dot{\underset{\sim}{\omega}}_1^r \times {}^{1'}\underset{\sim}{\Delta}_1 + {}^{1'}\underset{\sim}{\omega}_1^r \times ({}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{\Delta}_1) + 2({}^{1'}\underset{\sim}{\omega}_1^r \times {}^{1'}\underset{\sim}{\Delta}_1) + {}^{1'}\ddot{\underset{\sim}{\Delta}}_1\right] \quad (3\text{-}6h) $$

where:

$$ {}^{1'}\mathring{\underset{\sim}{\phi}}_1 = \left[{}^{1'}\mathring{\phi}_{in_1x} \quad {}^{1'}\mathring{\phi}_{in_1y} \quad {}^{1'}\mathring{\phi}_{in_1z}\right]^T $$

$$ {}^{1'}\underset{\sim}{\Delta}_1 = \left[\begin{array}{ccc} 0 & {}^{1'}\Delta_{in_1y} & {}^{1'}\Delta_{in_1z} \end{array}\right]^T $$

${}^{1'}\underset{\sim}{p}_1^{r\bullet}$ = vector from origin of $(x_{1-1}, y_{1-1}, z_{1-1})$ to origin of

$$(x'_1, y'_1, z'_1)$$

$$= \begin{bmatrix} a_1 & d_1 \sin\alpha_1 & d_1 \cos\alpha_1 \end{bmatrix}^T$$

$$\underset{\sim}{z}_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$E_1^{rf} = \text{transpose of } E_1^{fr}$$

$(\dot{:})$ indicates differentiation with respect to the primed (rigid) coordinate system of the link.

Velocities and accelerations of centers of mass of the elements are also required. These are described by the following equations:

$$^{i'}\underset{\sim}{\hat{\omega}}^r_{ij} = {}^{i'}\underset{\sim}{\omega}^r_i \qquad (3\text{-}7a)$$

$$^{i'}\underset{\sim}{\dot{\hat{\omega}}}^r_{ij} = {}^{i'}\underset{\sim}{\dot{\omega}}^r_i \qquad (3\text{-}7b)$$

$$^{i'}\underset{\sim}{\dot{\hat{v}}}^r_{ij} = {}^{i'}A_{i-1}\left[{}^{i-1}\underset{\sim}{\dot{v}}_{i-1}\right] + {}^{i'}\underset{\sim}{\dot{\hat{\omega}}}^r_{ij} \times \left[{}^{i'}p^{r*}_{i-1,ij} + {}^{i'}\hat{s}_{ij} \right]$$

$$+ {}^{i'}\underset{\sim}{\hat{\omega}}^r_{ij} \times \left[{}^{i'}\underset{\sim}{\hat{\omega}}^r_{ij} \times \left[{}^{i'}p^{r*}_{i-1,ij} + {}^{i'}\underset{\sim}{\hat{s}}_{ij} \right] \right] \qquad (3\text{-}7c)$$

In the recursive algorithm, these equations would be calculated on the forward recursion from the manipulator base to its tip.

3.3 Boundary Conditions.

The first element of each link is constrained such that it has

zero displacement with respect to the previous link at their common joint. The angle that this element makes with the previous link is the joint angle q_i(radians). Hence, u_{i0y}, u_{i0z}, φ_{i0x}, φ_{i0y}, and φ_{i0x} are all equal to zero. These conditions effectively clamp the i,th link at its joint with the $(i-1)^{'th}$ link so that the only relative motion between the two links at this point is the joint rotation. This boundary condition seems to represent the actual boundary conditions quite accurately, as verified by Hastings [8]. It seems that this phenomenon is due mainly to the presence of static friction in the joint's drive mechanism.

The boundary condition at the distal end of a link is not explicitly specified. Instead, motion of this end is indirectly determined by the effect of the following links on it. It will be shown later that the calculated mode shapes turn out to be almost the same as mode shapes calculated analytically under the assumption that the link's proximal end is clamped and its distal end is free. Modal frequencies change dramatically as the masses of following links vary or as the joint configuration of the manipulator changes, but the mode shapes remain the same!

3.4. Kinetic Energy Computation

As noted in chapter 1, we specialize our discussion somewhat to manipulators with links that have high aspect ratios. For such links, the most significant forms of vibration are flexure in both the y and z directions, and torsion about the longitudinal x-axis (see Figure 3.2). Longitudinal link-vibrations are usually of very small amplitudes and high frequencies and are hence neglected.

Figure 3.2 shows the vector displacement, r_{ij}, from the

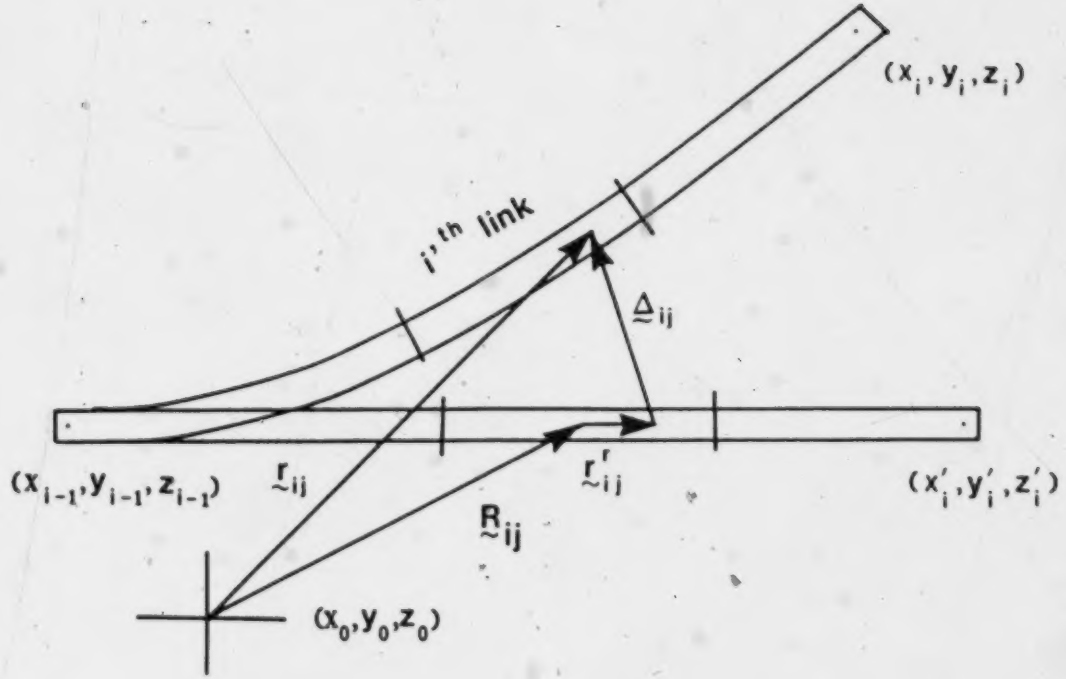


Figure 3.2. Definition of Vectors Used in Kinetic Energy Computation.

inertial reference frame (x_0, y_0, z_0) located at the manipulator base, to an arbitrary point on the j,th element of the i,th link. This vector is assumed to be measured with respect to the inertial reference frame (we shall omit left superscripts to indicate this). r_{ij} can be written in terms of the absolute displacement, R_{ij}, of the element's undeformed center-of-mass, and displacements due to flexibility, as follows:

$$\underset{\sim}{r}_{ij} = \underset{\sim}{R}_{ij} + \underset{\sim}{r}^{r}_{ij} + \underset{\sim}{\Delta}_{ij} \qquad (3\text{-}7)$$

where all terms are as defined in Figure 3.2. The kinetic energy of the element is given by:

$$KE_{ij} = \frac{1}{2}\int \dot{\underset{\sim}{r}}_{ij} \cdot \dot{\underset{\sim}{r}}_{ij}\, dm_{ij} + \frac{1}{2}\int \phi^{2}_{ijx}\, dm_{ij} \qquad (3\text{-}8)$$

where the integration is performed over the j'^{th} element if the i'^{th} link. Differentiating (3-7) yields

$$\dot{\underset{\sim}{r}}_{ij}(x'_{ij}) = \dot{\underset{\sim}{R}}_{ij} + \left[\tilde{\underset{\sim}{\omega}}^{r}_{ij} \times \underset{\sim}{r}^{r}_{ij}(x'_{ij})\right] + \tilde{\underset{\sim}{\omega}}^{r}_{ij} \times \underset{\sim}{\Delta}_{ij}(x'_{ij}) + \dot{\underset{\sim}{\Delta}}_{ij}(x'_{ij}) \qquad (3\text{-}9)$$

where:

$$\mathring{\underset{\sim}{\Delta}}_{ij}(x'_{ij}) = \begin{bmatrix} 0 \\ \mathring{\Delta}_{ijy}(x'_{ij}) \\ \mathring{\Delta}_{ijz}(x'_{ij}) \end{bmatrix}$$

After substituting (3-9) into (3-8) and performing the indicated

integration, the kinetic energy can be written as follows:

$$KE_{ij} = \frac{1}{2} m_{ij} \hat{\underset{\sim}{v}}^{r}_{ij} \cdot \hat{\underset{\sim}{v}}^{r}_{ij} + \frac{1}{2} \hat{\underset{\sim}{\omega}}^{r}_{ij} \cdot I^{f}_{ij} \hat{\underset{\sim}{\omega}}^{r}_{ij}$$

$$+ \hat{\underset{\sim}{v}}^{r}_{ij} \cdot \left[\left(\hat{\underset{\sim}{\omega}}^{r}_{ij} \times \hat{\underset{\sim}{\Delta}}_{ij} \right) + \mathring{\hat{\underset{\sim}{\Delta}}}_{ij} \right] + \hat{\underset{\sim}{\omega}}^{r}_{ij} \cdot \left[\mathring{\underset{\sim}{h}}_{ij} + \mathring{\underset{\sim}{f}}_{ij} \right]$$

$$+ \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijy} H^{ff}_{ijyy} \underset{\sim}{\vartheta}_{ijy} + \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijz} H^{ff}_{ijzz} \underset{\sim}{\vartheta}_{ijz}$$

$$+ \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijx} H^{ff}_{ijxx} \underset{\sim}{\vartheta}_{ijx} \qquad (3\text{-}10)$$

where:

$$\hat{\underset{\sim}{\Delta}}_{ij} = \left[0 \quad \hat{p}^{T}_{ijy} \underset{\sim}{\nu}_{ijy} \quad \hat{p}^{T}_{ijz} \underset{\sim}{\nu}_{ijz} \right]^{T}$$

$$\hat{p}_{ijz} = \hat{p}_{ijy} = \int p_{ijy}(x)\, dm_{ij} = m_{ij} \left[\frac{1}{2} \quad \frac{\ell_{ij}}{12} \quad \frac{1}{2} \quad -\frac{\ell_{ij}}{12} \right]$$

m_{ij} = mass of $ij^{'th}$ element

$$I^{f}_{ij} = \text{Trace}\left[J^{f}_{ij} \right] - J^{f}_{ij}$$

$$J^{f}_{ij} = J^{r}_{ij} + J^{ff}_{ij} + J^{rf}_{ij} + \left[J^{rf}_{ij} \right]^{T}$$

$$J^{ff}_{ij} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \underset{\sim}{\nu}^{T}_{ijy} H^{ff}_{ijyy} \underset{\sim}{\nu}_{ijy} & \underset{\sim}{\nu}^{T}_{ijy} H^{ff}_{ijyz} \underset{\sim}{\nu}_{ijz} \\ 0 & \underset{\sim}{\nu}^{T}_{ijz} H^{ff}_{ijzy} \underset{\sim}{\nu}_{ijy} & \underset{\sim}{\nu}^{T}_{ijz} H^{ff}_{ijzz} \underset{\sim}{\nu}_{ijz} \end{bmatrix}$$

$$J_{ij}^{rf} = \begin{bmatrix} 0 & J_{ijxy}^{rf}\, \underset{\sim}{v}_{ijy} & J_{ijxz}^{rf}\, \underset{\sim}{v}_{ijz} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$H_{ijyy}^{ff} = H_{ijzz}^{ff} = H_{ijyz}^{ff} = H_{ijzy}^{ff}$$

$$= \frac{m_{ij}}{420} \begin{bmatrix} 156 & 221_{ij} & 54 & -131_{ij} \\ 221_{ij} & 41_{ij}^2 & 131_{ij} & -31_{ij}^2 \\ 54 & 131_{ij} & 156 & -221_{ij} \\ -131_{ij} & -31_{ij}^2 & -221_{ij} & 41_{ij}^2 \end{bmatrix}$$

$$J_{ijxy}^{rf} = J_{ijxz}^{rf} = \frac{m_{ij}}{10} \begin{bmatrix} -1 & -\frac{l_{ij}}{12} & 1 & -\frac{l_{ij}}{12} \end{bmatrix}$$

$$\overset{\circ}{\underset{\sim}{h}}_{ij} = \begin{bmatrix} 0 \\ -J_{ijxz}^{rf}\, \underset{\sim}{\theta}_{ijy} \\ J_{ijxy}^{rf}\, \underset{\sim}{\theta}_{ijz} \end{bmatrix}$$

$$\overset{\circ}{\underset{\sim}{f}}_{ij} = \begin{bmatrix} V_{ijy}^T\, H_{ijyz}^{ff}\, \underset{\sim}{\theta}_{ijz} - \underset{\sim}{v}_{ijz}^T\, H_{ijzy}^{ff}\, \underset{\sim}{\theta}_{ijy} \\ 0 \\ 0 \end{bmatrix}$$

$$H^{ff}_{ijxx} = m_{ij}\begin{bmatrix} \frac{1}{3} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{3} \end{bmatrix}$$

3.5. Potential Energy Computation.

The elastic potential energy computation follows the traditional procedure as described in [6], and can be summarized by the following equation:

$$P^{e}_{ij} = \frac{1}{2} \underset{\sim}{v}^{T}_{ijy} K_{ijy} \underset{\sim}{v}^{T}_{ijy} + \frac{1}{2} \underset{\sim}{v}^{T}_{ijz} K_{ijz} \underset{\sim}{v}^{T}_{ijz} + \frac{1}{2} \underset{\sim}{v}^{T}_{ijx} K_{ijx} \underset{\sim}{v}^{T}_{ijx} \qquad (3\text{-}11)$$

where:

$$K_{ijy} = \frac{E_{ijy} I_{ijy}}{l^{3}_{ij}} \bar{K}_{ijy}$$

$$K_{ijz} = \frac{E_{ijz} I_{ijz}}{l^{3}_{ij}} \bar{K}_{ijz}$$

$$K_{ijx} = \frac{G_{ijx} J_{ijx}}{l_{ij}} \bar{K}_{ijx}$$

$$\bar{K}_{ijy} = \bar{K}_{ijz} = \frac{m_{ij}}{420} \begin{bmatrix} 12 & 61_{ij} & -12 & 61_{ij} \\ 61_{ij} & 41_{ij}^2 & -61_{ij} & 21_{ij}^2 \\ -12 & -61_{ij} & 12 & -61_{ij} \\ 61_{ij} & 21_{ij}^2 & -61_{ij} & 41_{ij}^2 \end{bmatrix}$$

$$\bar{K}_{ijx} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

The total elastic potential energy, P^e, is the sum all P^e_{ij} over all elements and all links.

3.6. Inverse Dynamic Equations.

Derivation of the joint equations of motion involve the expansion of the derivatives indicated in the following Lagrangian Equations:

$$\tau_i = \frac{d}{dt}\left[\frac{\partial K}{\partial \dot{q}_i}\right] - \frac{\partial K}{\partial q_i} + \frac{\partial P_g}{\partial *} \qquad (3\text{-}12a)$$

$$\tau_* = \frac{d}{dt}\left[\frac{\partial K}{\partial \dot{*}}\right] - \frac{\partial K}{\partial *} + \frac{\partial P_e}{\partial *} + \frac{\partial P_g}{\partial *} \qquad (3\text{-}12b)$$

where:

$$K = \sum_{i=1}^{N} \sum_{j=1}^{n_i} K_{ij} \; ; \qquad N = \text{total number of links.}$$

P_e is the total elastic potential energy,

P_g is the total gravitational potential energy,

"*" indicates one of u_{1ky}, u_{1kz}, φ_{1kx}, φ_{1ky}, or φ_{1kx} and $\dot{*}$ indicates its time derivative.

A procedure similar to that described in the previous chapter is followed. The following identities are used in the simplification of the expressions that result from the expansions:

$$\frac{d}{dt}\left[\frac{\partial \hat{\underset{\sim}{v}}^r_{ij}}{\partial \dot{q}_1}\right] = \frac{\partial \hat{\underset{\sim}{v}}^r_{ij}}{\partial q_1} \qquad (3\text{-}13a)$$

$$\frac{d}{dt}\left[\frac{\partial \underset{\sim}{\omega}^r_{ij}}{\partial \dot{q}_1}\right] = \frac{\partial \underset{\sim}{\omega}^r_{ij}}{\partial q_1} + \underset{\sim}{\omega}^r_{ij} \times \frac{\partial \underset{\sim}{\omega}^r_{ij}}{\partial \dot{q}_1} \qquad (3\text{-}13b)$$

$$\left[\frac{\partial\ ?}{\partial q_1}\right] = \frac{\partial\ ?}{\partial \dot{q}_1} \times ? \qquad (3\text{-}13c)$$

$$\left[\frac{\partial\ ?}{\partial *}\right] = \frac{\partial\ ?}{\partial \dot{*}} \times ? \qquad (3\text{-}13d)$$

where ? can be replaced in (3-13c) and (3-13d) by any of $\hat{\underset{\sim}{A}}_{ij}$, $\dot{\hat{\underset{\sim}{A}}}_{ij}$, $\overset{\circ}{\underset{\sim}{h}}_{ij}$ or $\overset{\circ}{\underset{\sim}{f}}_{ij}$, and * and $\dot{*}$ are as in (3-12c). Proofs of these identities are similar to proofs of the corresponding identities in chapter 2 and will therefore not be give here.

After simplification, the joint equations of motion become:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{q}_1}\right] - \frac{\partial K}{\partial q_1} =$$

$$\sum_{i=1}^{N}\sum_{j=1}^{n_1}\left\{\left[\frac{\partial\, \hat{\underset{\sim}{v}}^{r}_{ij}}{\partial\, \dot{q}_1}\right] \cdot F_{ij} + \left[\frac{\partial\, \hat{\underset{\sim}{\omega}}^{r}_{ij}}{\partial\, \dot{q}_1}\right] \cdot N_{ij}\right\} \qquad (3\text{-}14)$$

where:

$$\underset{\sim}{F}_{ij} = m_{ij}\, \hat{\underset{\sim}{v}}^{r}_{ij} + \dot{\underset{\sim}{\omega}}^{r}_{ij} \times \hat{\underset{\sim}{\Delta}}_{ij} + \underset{\sim}{\omega}^{r}_{ij} \times (\underset{\sim}{\omega}^{r}_{ij} \times \hat{\underset{\sim}{\Delta}}_{ij}) + 2\underset{\sim}{\omega}^{r}_{ij} \times \hat{\dot{\underset{\sim}{\Delta}}}_{ij} + \hat{\ddot{\underset{\sim}{\Delta}}}_{ij}$$

$$\underset{\sim}{N}_{ij} = I^{f}_{ij}\, \hat{\underset{\sim}{\omega}}^{r}_{ij} + \mathring{I}^{f}_{ij}\, \hat{\underset{\sim}{\omega}}^{r}_{ij} + \hat{\underset{\sim}{\omega}}^{r}_{ij} \times I^{f}_{ij}\, \hat{\underset{\sim}{\omega}}^{r}_{ij} + \hat{\Delta}_{ij} \times \hat{v}^{r}_{ij} + \hat{\underset{\sim}{\omega}}^{r}_{ij} \times (\mathring{\underset{\sim}{h}}_{ij} + \mathring{\underset{\sim}{f}}_{ij}) + \ddot{\underset{\sim}{h}}_{ij} + \ddot{\underset{\sim}{f}}_{ij}$$

The only form of potential energy that is dependent on q_j is the gravitational potential energy. Its effect is included in the same manner as in Chapter 2. That is, the manipulator-base is given an acceleration equal to the gravitational acceleration.

The flexibility equation for each element is slightly more complicated in that there are "extra" terms in the equation which are dependent only on the link under consideration. After simplification we have:

$$\frac{d}{dt}\left[\frac{\partial K}{\partial \dot{*}}\right] - \frac{\partial K}{\partial *} =$$

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i} \left\{ \left[\frac{\partial \hat{\underset{\sim}{V}}^{r}_{ij}}{\partial \dot{\bullet}} \right] \cdot \underset{\sim}{F}_{ij} + \left[\frac{\partial \underset{\sim}{\omega}^{r}_{ij}}{\partial \dot{\bullet}} \right] \cdot \underset{\sim}{N}_{ij} \right\} + t'_{\bullet} \quad (3\text{-}15)$$

where:

$$t'_{\bullet} = \sum_{j=1}^{n_i} \left\{ \hat{\underset{\sim}{V}}^{r}_{ij} \cdot \frac{\partial \overset{\circ}{\underset{\sim}{\Delta}}_{ij}}{\partial \bullet} + \hat{\underset{\sim}{\omega}}^{r}_{ij} \cdot \left[\frac{\partial \overset{\circ}{\underset{\sim}{h}}_{ij}}{\partial \bullet} + \frac{\partial \overset{s}{\underset{\sim}{f}}_{ij}}{\partial \bullet} \right] \right.$$

$$+ \hat{\underset{\sim}{\omega}}^{r}_{ij} \cdot \left[\frac{d}{dt} \left[\frac{\partial \overset{s}{\underset{\sim}{f}}_{ij}}{\partial \dot{\bullet}} \right] + \frac{\partial \overset{s}{\underset{\sim}{f}}_{ij}}{\partial \bullet} \right] + \frac{d}{dt} \left[\frac{\partial \overset{\circ}{b}_{ij}}{\partial \dot{\bullet}} \right]$$

$$\left. - \frac{1}{2} \hat{\underset{\sim}{\omega}}^{r}_{ij} \cdot \frac{\partial I^{f}_{ij}}{\partial \bullet} \hat{\underset{\sim}{\omega}}^{r}_{ij} \right\} \quad (3\text{-}16)$$

where "•" is as in (3-13), and:

$$\overset{\circ}{b}_{ij} = \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijy} H^{ff}_{ijyy} \underset{\sim}{\vartheta}_{ijy} + \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijz} H^{ff}_{ijzz} \underset{\sim}{\vartheta}_{ijz}$$

$$+ \frac{1}{2} \underset{\sim}{\vartheta}^{T}_{ijx} H^{ff}_{ijxx} \underset{\sim}{\vartheta}_{ijx}$$

The terms that comprise $t'_{\bullet}$ are not included in the summation over the links (represented by the summation over 'i') in equation (3-15) are the "extra" terms. They depend on the j'^{th} and $(j+1)'^{st}$ elements of the i'^{th} link only and are quite straightforward to compute. For example:

$$\sum_{j=1}^{n_i} \left\{ \hat{\underset{\sim}{V}}^{r}_{ij} \cdot \frac{\partial \overset{\circ}{\underset{\sim}{\Delta}}_{ij}}{\partial \dot{u}_{iky}} \right\} = \hat{\underset{\sim}{V}}^{r}_{ik} \cdot \left[0 \quad \hat{p}_{iky}[3] \quad 0 \right]$$

$$+ \overset{\circ r}{\underset{\sim}{V}}_{1(k+1)} \cdot \begin{bmatrix} 0 & \hat{p}_{1(k+1)y}[1] & 0 \end{bmatrix}$$

The other terms are just as straightforward.

Elastic potential energy effects are included as follows:

$$\frac{\partial P^*}{\partial u_{1ky}} = K_{1ky}[3;] \underset{\sim}{V}_{1ky} + K_{1(k+1)y}[1;] \underset{\sim}{V}_{1(k+1)y}$$

A similar expression holds for $\frac{\partial P^*}{\partial u_{1kz}}$. Expressions for $\frac{\partial P^*}{\partial \varphi_{1ky}}$ and $\frac{\partial P^*}{\partial \varphi_{1kz}}$ are of the form:

$$\frac{\partial P^*}{\partial \varphi_{1ky}} = K_{1ky}[4;] \underset{\sim}{V}_{1ky} + K_{1(k+1)y}[2;] \underset{\sim}{V}_{1(k+1)y}$$

and finally:

$$\frac{\partial P^*}{\partial \varphi_{1kx}} = K_{1kx}[2;] \underset{\sim}{V}_{1kx} + K_{1(k+1)x}[1;] V_{1(k+1)x}$$

The potential energy term can be combined with t'_* to yield:

$$t_* = t'_* + \frac{\partial P^*}{\partial *}$$

3.7. Recursive Algorithm for Inverse Dynamics Calculation.

The main advantage of this formulation of flexible manipulator dynamics is its relative computational efficiency. This efficiency results from two characteristics: (1) the recursive nature of the formulation, and (2) the representation of rotational kinematics in terms of angular velocities, $\underset{\sim}{\omega}$ and $\overset{\circ}{\underset{\sim}{\phi}}$, instead of derivatives of transformation matrices. Together, these factors cause a substitution of storage-space for computation time. More terms can

now be calculated once and stored for use in various parts of the computational algorithm. The recursive nature of equations (3-14) and (3-15) is not immediately obvious. Further simplification is necessary before they can be written in their final, recursive form. This is accomplished by making use of the following identities:

$$\frac{\partial \hat{\omega}^r_{ij}}{\partial \dot{q}_l} = \underset{\sim}{z}_{l-1}; \qquad i \geq j;$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{q}_l} = \underset{\sim}{z}_{l-1} \times \hat{p}_{l-1;ij}; \qquad i \geq l;$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{u}_{lky}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{u}_{lkz}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{\varphi}_{lkx}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \times \hat{p}_{l;ij} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{\varphi}_{lky}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \hat{p}_{l;ij} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \hat{v}^r_{ij}}{\partial \dot{\varphi}_{lkz}} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \times \hat{p}_{l;ij} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \omega^r_{ij}}{\partial \dot{\varphi}_{lkx}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ for } k = n_l,\ i > l,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \omega_{ij}^{r}}{\partial \dot{\varphi}_{1ky}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{for } k = n_1,\ i > 1,\ \underset{\sim}{0} \text{ otherwise;}$$

$$\frac{\partial \omega_{ij}^{r}}{\partial \dot{\varphi}_{1kz}} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \quad \text{for } k = n_1,\ i > 1,\ \underset{\sim}{0} \text{ otherwise;}$$

The resulting algorithm can now be written in the form of a recursion from the manipulator tip down to its base. Define the following:

$$^{1'}\underset{\sim}{f}_{1+1} = E_1^{fr}\ {}^{1}\underset{\sim}{f}_{1+1} \tag{3-17a}$$

$$^{1'}\underset{\sim}{n}_{1+1} = E_1^{fr}\ {}^{1}\underset{\sim}{n}_{1+1} \tag{3-17b}$$

$$^{1'}\underset{\sim}{N}_1 = \sum_{j=1}^{n_1} {}^{1'}\underset{\sim}{N}_{1j} \tag{3-18a}$$

$$^{1'}\underset{\sim}{F}_1 = \sum_{j=1}^{n_1} {}^{1'}\underset{\sim}{F}_{1j} \tag{3-18b}$$

$$^{(1-1)}\underset{\sim}{f}_1 = {}^{(1-1)}A_1\left[{}^{1'}\underset{\sim}{F}_1 + {}^{1'}\underset{\sim}{f}_{1+1} \right] \tag{3-19}$$

$$^{(1-1)}\underset{\sim}{n}_1 = {}^{(1-1)}A_1\left[{}^{1'}\underset{\sim}{N}_1 + \sum_{j=1}^{n_1}\left({}^{1'}\hat{\underset{\sim}{p}}_{1-1;1j} \times {}^{1'}\underset{\sim}{F}_{1j} \right) \right]$$

$$+ \; {}^{1'}n_{l+1} + \left[{}^{1'}p_l^{r^*} + {}^{1'}\Delta_l \right] \times {}^{1'}f_{l+1} \Big] \qquad (3\text{-}20)$$

Then, inverse dynamics due to flexibility effects are as follows:

$$\tau_{u_{lky}} = \begin{cases} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cdot {}^{1'}f_{l+1} + t_{u_{lky}}; & k = n_l \\ t_{u_{lky}}; & k < n_l \end{cases} \qquad (3\text{-}21)$$

$$\tau_{u_{lkz}} = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot {}^{1'}f_{l+1} + t_{u_{lkz}}; & k = n_l \\ t_{u_{lkz}}; & k < n_l \end{cases} \qquad (3\text{-}22)$$

$$\tau_{\varphi_{lkx}} = \begin{cases} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \cdot {}^{1'}n_{l+1} + t_{\varphi_{lkx}}; & k = n_l \\ t_{\varphi_{lkx}}; & k < n_l \end{cases} \qquad (3\text{-}23)$$

$$\tau_{\varphi_{lky}} = \begin{cases} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot {}^{1'}n_{l+1} + t_{\varphi_{lky}}; & k = n_l \\ t_{\varphi_{lky}}; & k < n_l \end{cases} \qquad (3\text{-}24)$$

$$\tau_{\varphi_{1kz}} = \begin{cases} \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \cdot {}^{1'}n_{1+1} + t_{\varphi_{1kz}}; & k = n_1 \\ t_{\varphi_{1kz}}; & k < n_1 \end{cases} \qquad (3\text{-}25)$$

The joint torques are calculated by the following equation:

$$\tau_1 = \text{z-component of } {}^{1-1}n_1 \qquad (3\text{-}26)$$

In an actual implementation of this algorithm, ${}^{1'}N_1$, ${}^{1'}F_1$, and ${}^{1'}\hat{P}_{1-1;1j} \times {}^{1'}F_{1j}$ are calculated on a forward recursion from the manipulator base to its tip and stored. Then, starting with ${}^{N}f_{N+1}$ and ${}^{N}n_{N+1}$, which are forces and torques exerted on the manipulator by its payload, equations (3-16) to (3-25) are calculated recursively back to the base. Because of the similarity of this algorithm to the Newton-Euler algorithm of Luh et al [43], we shall refer to the algorithm presented in this section as the "Newton-Euler Finite-Element" (NEFE) method for flexible manipulators.

It is sometimes convenient to simulate the effect of a concentrated payload on the dynamic response of a flexible manipulator. In such a case, the backward recursion is started by setting ${}^{N}n_{N+1}$ to the zero vector and:

$${}^{N}f_{n+1} = M_L \, {}^{N}\dot{V}_N .$$

3.8. Recursive Algorithm for Calculation of Forward Dynamics.

The forward dynamics of a flexible manipulator are, as before, represented by the following equation:

$$H \ddot{\underset{\sim}{z}} = \underset{\sim}{R} \qquad (3\text{-}27)$$

where:

$\underset{\sim}{z}$ = the vector of all generalized coordinates

$= \left[q_1, q_2, \cdots q_n, u_{11y}, u_{11x}, \varphi_{11x}, \varphi_{11y}, \varphi_{11z}, \cdots\right]$

H = the symmetric Generalized Inertia Matrix (GIM)

$\underset{\sim}{R}$ = vector of remaining dynamics and external forcing terms.

As pointed out in Chapter 2, numerical simulation of a flexible manipulator is accomplished by solving (3-27) for accelerations and integrating twice to yield velocities and position variables. The vector $\underset{\sim}{R}$ is the difference between the external torques applied at the manipulator joints, and the inverse dynamics obtained by setting all accelerations to zero.

To calculate H, we again notice that each column is equal to the inverse dynamics obtained when all references to gravity, generalized velocities, and elastic forces are omitted from the algorithm, and the generalized acceleration vector set to zero except for a "1" in the location corresponding to that particular column. It is computationally expensive to actually follow this procedure in a computer program. Instead, the inverse dynamic equations themselves are modified to reflect the effect of the procedure. The result is a recursive algorithm for directly calculating the terms of the GIM. When calculating the column corresponding to a particular element or joint generalized variable, the rest of the manipulator from this element or joint

up to the manipulator tip is considered to be frozen in its deformed state at that given instant of time. The second derivative of this generalized variable is set to unity in the inverse dynamic equations, resulting in much simpler expressions. A run of this simpler algorithm from the element in question down to the manipulator base yields the portion of the column that lies in the upper triangle. Since H is symmetrical, this is all that is needed. Starting with the last element of the last link and moving down to the base, the entire upper triangle of the GIM is calculated.

To describe the forward dynamics algorithm, we shall use the following notation for terms of the H matrix:

$h_{i:j}$ = the term in the column corresponding to $\ddot{q}_j$ and the row corresponding to $\ddot{q}_i$;

$h_{i:*1k?}$ = the term in the column corresponding to $\ddot{*}_{1k?}$ and the row corresponding to $\ddot{q}_i$

$h_{*1k?:j}$ = the term in the column corresponding to $\ddot{q}_j$ and the row corresponding to $\ddot{*}_{1k?}$;

$h_{*1k?:*1j?}$ = the term in the column corresponding to $\ddot{*}_{1j?}$ and the row corresponding to $\ddot{*}_{1k?}$;

where * refers to one of u or φ and ? is either x, y ar z. In the algorithm that follows, we assume that the manipulator payload is a concentrated mass. Then, starting with:

$$M_{N+1} = \text{payload mass.} \tag{3-28}$$

$$\overset{N'}{\underset{\sim}{c}}_{N+1} = \underset{\sim}{0} \tag{3-29}$$

$$ {}^{N'}\underset{\sim}{\mathcal{G}}_{N+1} = [0] \qquad (3\text{-}30) $$

calculate the following for the j'th link during the recursion down to the manipulator base:

$$ M_j = M_{j+1} + m_j, \quad (\text{where } m_j = \sum_k m_{jk}) \qquad (3\text{-}31) $$

$$ M_j\, {}^{j'}\underset{\sim}{c}_j = M_{j+1}\left[{}^{j'}\underset{\sim}{p}_j^{r*} + {}^{j'}\underset{\sim}{\Delta}_j + {}^{j'}\underset{\sim}{c}_{j+1} \right] + \sum_{k=1}^{n_j} m_{jk}\left[{}^{j'}\underset{\sim}{p}_{j-1;jk} + {}^{j'}\hat{\underset{\sim}{s}}_{jk} \right] + {}^{j'}\hat{\underset{\sim}{\Delta}}_{jk} \Bigg] \Bigg\} \qquad (3\text{-}32) $$

$$ {}^{(j-1)'}\underset{\sim}{c}_j = E_{j-1}^{fr}\; {}^{(j-1)}A_j\; {}^{j'}\underset{\sim}{c}_j\; {}^{j'}A_{j-1}\; E_{j-1}^{rf} \qquad (3\text{-}33) $$

$$ \underset{\sim}{X}_1 = {}^{j'}\underset{\sim}{c}_{j+1} + {}^{j'}\underset{\sim}{p}_j^{r*} + {}^{j'}\underset{\sim}{c}_j \qquad (3\text{-}34) $$

$$ \underset{\sim}{X}_2 = {}^{j'}\underset{\sim}{p}_{j-1,jk} + {}^{j'}\hat{\underset{\sim}{s}}_{jk} + \frac{{}^{j'}\hat{\underset{\sim}{\Delta}}_{jk}}{m_{jk}} - {}^{j'}\underset{\sim}{c}_j \qquad (3\text{-}35) $$

$$ {}^{(j-1)'}\underset{\sim}{\mathcal{G}}_j = E_{j-1}^{fr}\; {}^{j-1}A_{j'} \left\{ {}^{j'}\underset{\sim}{\mathcal{G}}_{j+1} + M_{j+1}\, F(\underset{\sim}{X}_1) + \sum_{k=1}^{n_j} \left\{ {}^{j'}I_{jk} - F({}^{j'}\hat{\underset{\sim}{\Delta}}_{jk})/m_{jk} + m_{jk}\, F(\underset{\sim}{X}_{2k}) \right\} \right\} {}^{j'}A_{j-1} E_{j-1}^{rf} \qquad (3\text{-}36) $$

where $F(\underset{\sim}{X}) = (\underset{\sim}{X} \cdot \underset{\sim}{X})I - \underset{\sim}{X}\,\underset{\sim}{X}^T$.

A column corresponding to the generalized variable u_{jky} is calculated by forming the following vectors:

$$ {}^{j'}\underset{\sim}{f}_{j+1:ujky} = M_{j+1}\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{if } k = n_j \qquad (3\text{-}37) $$

$$ = \underset{\sim}{0} \text{ otherwise;} $$

$$ {}^{j'}\underset{\sim}{n}_{j+1:ujky} = M_{j+1}\ {}^{j'}\underset{\sim}{c}_{j+1} \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{if } k = n_j \qquad (3\text{-}38) $$

$$ = \underset{\sim}{0} \qquad \text{otherwise;} $$

$$ {}^{j'}\underset{\sim}{f}_{j:ujky} = {}^{j'}\underset{\sim}{f}_{j+1:ujky} + \begin{bmatrix} 0 \\ \hat{p}_{jky}[3] \end{bmatrix} + \begin{bmatrix} 0 \\ \hat{p}_{j(k+1)y}[3] \\ 0 \end{bmatrix} \qquad (3\text{-}39) $$

$$ {}^{j'}\underset{\sim}{n}_{j:ujky} = {}^{j'}\underset{\sim}{n}_{j+1:ujky} + \left({}^{j'}p_j^{r^*} + {}^{j'}\underset{\sim}{\Delta}_j \right) \times {}^{j'}\underset{\sim}{f}_{j+1:ujky} + \left({}^{j'}p_j^{r^*} + {}^{j'}\hat{\underset{\sim}{s}}_{jk} \right) \times \begin{bmatrix} 0 \\ \hat{p}_{jky}[3] \\ 0 \end{bmatrix} $$

$$+ \left[{}^{j'}\underset{\sim}{p}_{j}^{r*} + {}^{j'}\underset{\sim}{\hat{s}}_{j(k+1)} \right] \times \begin{bmatrix} 0 \\ \hat{p}_{j(k+1)y}\,[1] \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} 0 \\ 0 \\ J^{rf}_{jkxy}[3] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ J^{rf}_{j(k+1)xy}[0] \end{bmatrix}$$

$$+ \begin{bmatrix} V^{T}_{jky}\, J_{jkyy}[;3] \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} V^{T}_{j(k+1)y}\, J_{j(k+1)yy}[;1] \\ 0 \\ 0 \end{bmatrix} \qquad (3\text{-}40)$$

For $i < j$:

$${}^{(i-1)'}\underset{\sim}{f}_{i:ujky} = {}^{i-1}A_{i}\,{}^{i'}\underset{\sim}{f}_{i+1:ujky} \qquad (3\text{-}41)$$

$${}^{(i-1)'}\underset{\sim}{n}_{i:ujky} = {}^{i-1}A_{i}\left\{ {}^{i'}\underset{\sim}{n}_{i+1:ujky} + \left[{}^{i'}\underset{\sim}{p}_{i}^{r*} + {}^{i'}\underset{\sim}{\Delta}_{i} \right] \times {}^{i'}\underset{\sim}{f}_{i+1:ujky} \right\} \qquad (3\text{-}42)$$

The terms of this column, from the diagonal term up to the first term, are obtained by substituting equations (3-36) and (3-37) into equations (3-20) to (3-25) and following through with the

recursive procedure.

By similar procedures, the columns corresponding to the other flexibility variables can be calculated.

A column corresponding to the joint variable q_j is found by first forming the vector:

$$ {}^{j-1}\underset{\sim}{f}_j = E_j^{rf}\, \underset{\sim}{z}_0 \times M_j\, {}^{j-1}\underset{\sim}{c}_j \qquad (3\text{-}43) $$

$$ {}^{j-1}\underset{\sim}{n}_j = {}^{j-1}g_j\, E_j^{rf}\, \underset{\sim}{z}_0 + {}^{j-1}\underset{\sim}{c}_j \times {}^{j-1}\underset{\sim}{f}_j \qquad (3\text{-}44) $$

where $\underset{\sim}{z}_0 = [0 \;\; 0 \;\; 1]^T$. The diagonal term is given by:

$$ h_{j,j} = \text{z-component of } (E_j^{rf}\, {}^{j-1}\underset{\sim}{n}_j) \qquad (3\text{-}45) $$

The rest of the elements are found by substituting equations (3-43) and (3-44) into (3-20) to (3-25) and again following through with the recursion.

This completes the recursive procedure for calculating the Generalized Inertia Matrix.

3.9. Simulation Example.

The forward dynamic algorithm is demonstrated by simulation of the two-link flexible manipulator shown in Figure 2.2. The manipulator dimensions are also given in Figure 2.2. Each link is divided into four elements. The initial values of the deflection variables are all set to zero, and the joint angles are set to q_1 = -80 degrees, and q_2 = 10 degrees. The arm is allowed to swing freely under gravity. Figure 3.3 shows the simulation results.

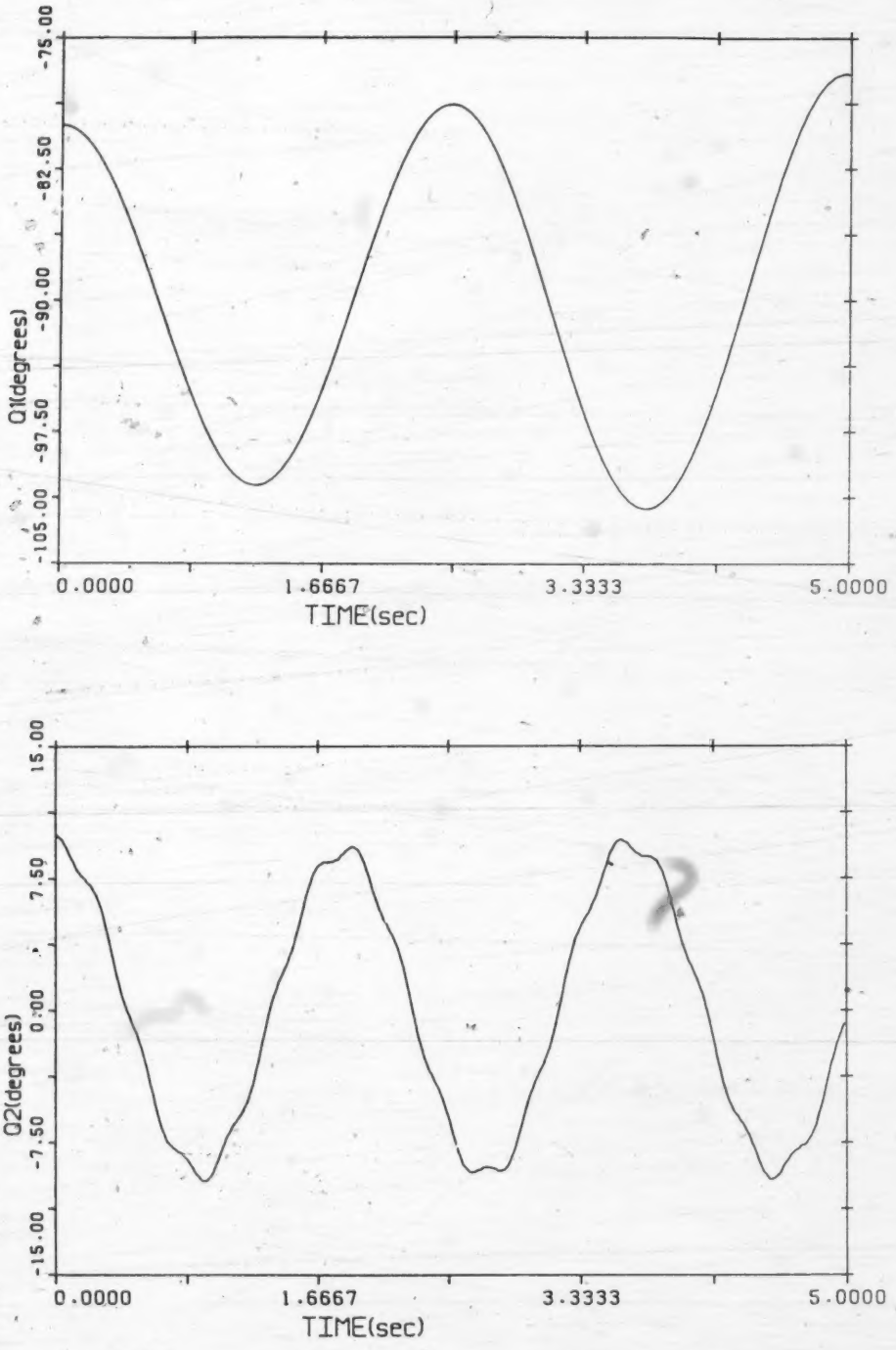
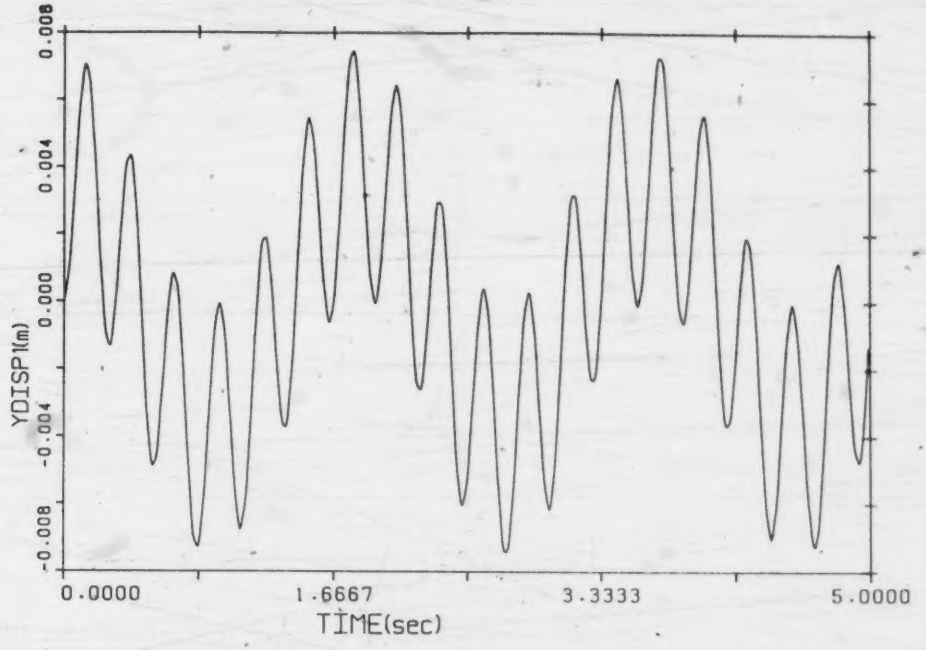


Figure 3.3 Results of the Finite Element Simulation of the Two-Link Manipulator shown in Figure 2.2. (Labels are explained in the text.)



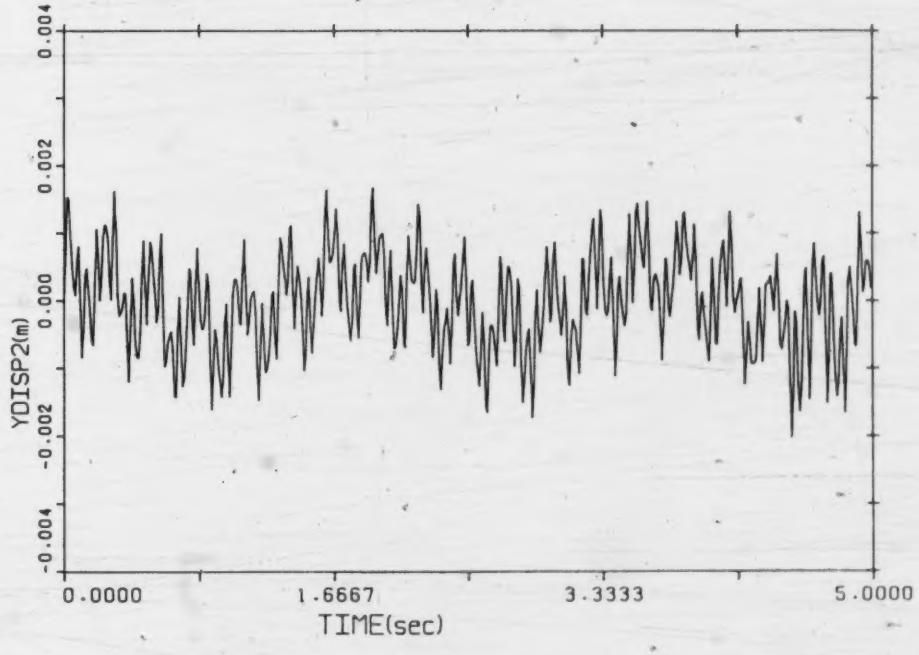
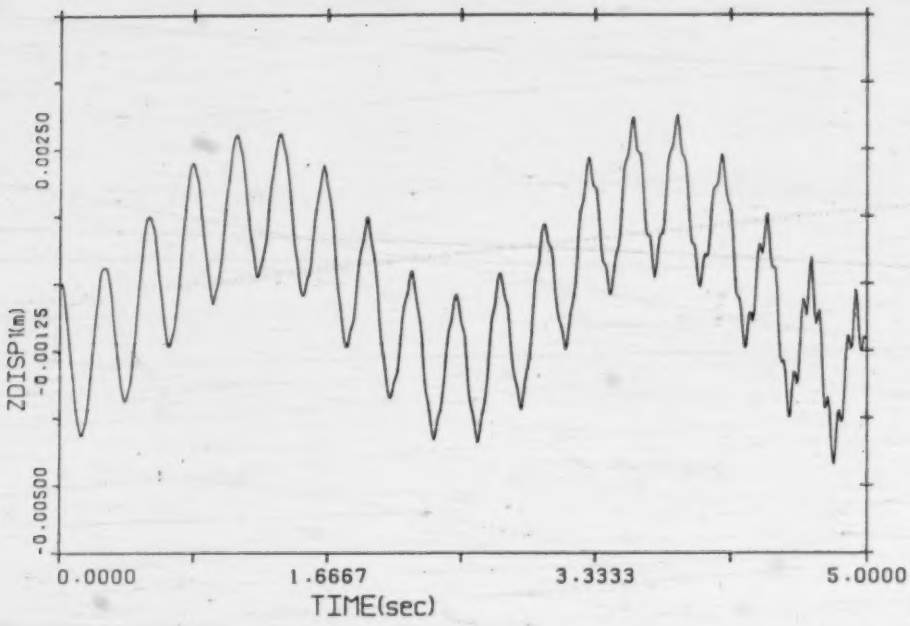


Figure 3.3. (Cont'd).



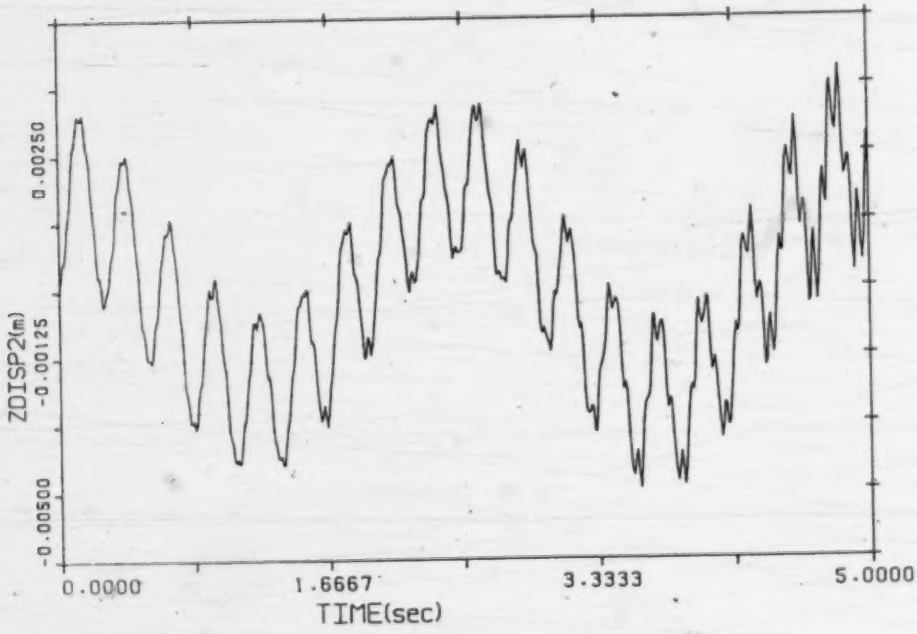


Figure 3.3. (Cont'd).

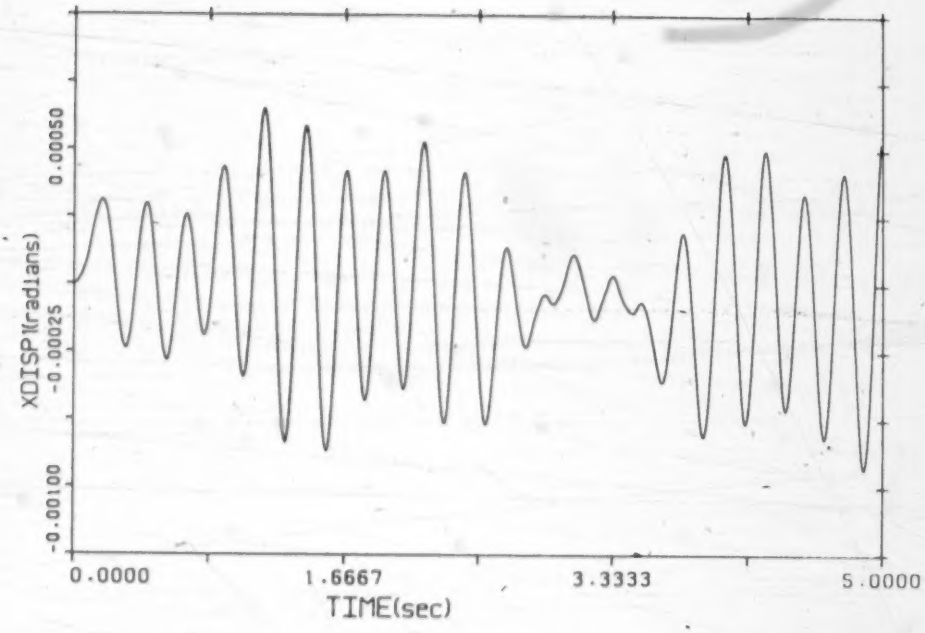


Figure 3.3. (Cont'd).

Plots of the gross joint motions (labeled "JOINT 1" and JOINT 2" in Figure 3.3), and the fine oscillatory vibrations of the tips (labeled "YDISP", "ZDISP" and "TORSION" in Figure 3.3) of each link are shown. Note that the fine oscillatory motions of the link tips appear superimposed onto the gross motions that result from joint motion. It is also worth noting that, as in Chapter 2, the frequency of the the flexible oscillations are higher than the frequency of the joint oscillations.

3.10. Comparison Between Finite Element and Assumed Modes Models.

In the simulation studies that have been reported in the literature, clamped-free mode-shapes for freely vibrating beams have been used as admissible functions approximating the actual mode shapes of flexible links [8,90]. There has been some doubt, however, as to whether these mode shapes satisfactorily describe link vibrations under the all the different boundary conditions that are imposed on a link as it moves in the chain of links. If the mode shapes are known then the deflection variables can be obtained from deflection measurements in a manner described by Nemir [90]. Nemir has also recently reported experimental results on a single link with an end-point mass. In his experiments, he found that clamped-free mode shapes were the most satisfactory. Both clamped-clamped and clamped-mass eigenfunctions were rejected.

There are noticeable differences between the simulation results shown in Figure 2.3, where link flexibility is modeled by the assumed modes method, and the simulation results shown in Figure 3.3, where link flexibility is modeled by the finite

element method. A major contributor to these differences is the inaccuracy of the finite element representation of the higher modes of vibration when only each link is divided into only four elements. However, another significant reason for these differences is that a very large number of iterations is needed to simulate five seconds of manipulator motion when the finite element method is used. In this method, a very small integration time step is required, since high vibrational frequencies (though inaccurate) are present. Numerical errors accumulate and can be noticed in the plots of Figure 3.3.

A more accurate comparison between the assumed modes and finite element methods can be performed by considering motion in only one plane. In this case, the number of elements that each link can be divided into can be increased without increasing the dimension of the overall system. In the comparison that follows, a two-link, planar manipulator, with both links flexible, is considered. Each link is divided into eight elements. Four modes per link are used in the assumed modes method.

In this section we compare mode shapes and modal frequencies that are estimated from finite element data with the clamped-free mode shapes and modal frequencies calculated by use of the assumed modes method described in Chapter 2. The mode-shapes and modal frequencies are estimated under several different "payload conditions" and for different configurations of the manipulator joints.

The mode shapes are calculated by partitioning the manipulator dynamic equation as given by equation (3-27) into two separate but coupled equations that represent the joint and flexibility motion

respectively. Only the inertia and stiffness terms are retained in the flexibility equation. The equation of flexibility then becomes:

$$H \ddot{\underset{\sim}{\delta}} = K \delta \qquad (3\text{-}46)$$

The matrices H and K are then partitioned into submatrices corresponding to each flexible link then further into matrices that correspond to each of the x, y and z directions. Each resulting equation is of the form of (3-46). For example, the equation describing the subsystem for the i'^{th} link in the y direction is as follows:

$$H_y \ddot{\underset{\sim}{\delta}}_y = K_y \underset{\sim}{\delta}_y \qquad (3\text{-}47)$$

Each equation represents a general eigenvalue problem which can be solved for modal frequencies and shape functions. The shape functions (eigenvectors) can then be used to calculate modal parameters that are needed in the assumed modes algorithms of chapter 2.

The procedure for forming equation (3-47) neglects the submatrices that represent coupling between the links and between the different directions associated with a particular link. This does not mean that interaction between the links is neglected. Interaction between the links is included in the elements that make up the matrix H_y in equation (3-47). The terms of the coupling submatrices are found to be quite small when compared with the elements of H_y. It seems that neglecting them does not

detract from the usefulness of equation (3-47) for calculating modal shapes and frequencies.

Figure 3.4 shows plots of the fundamental mode shape of deflection in the transverse y-direction of the inner link of a two-link planar manipulator. Each link is of rectangular cross-section with dimensions given in Table 3.1, and is divided into eight elements of equal length. The mass of the payload is varied from zero up to a value slightly greater than the combined mass of the two links. Mode shapes calculated by the finite element method are compared to the clamped-free modeshape given by the following expression:

$$\bar{\Delta}_y(x) = \cosh(\lambda x) - \cos(\lambda x) + \sigma \left[\sinh(\lambda x) - \sin(\lambda x) \right] \quad (3\text{-}48)$$

where, for the first mode:

$\sigma = 0.7341$

$\lambda = 1.8751/\ell$, and ℓ is the length of the link.

This expression is plotted in Figure 3.4, along with the mode shape of the first mode of the first link for five different cases which correspond to the second link having different masses. The five cases are summarized in Table 3.2. It is seen from the plots that the mode shapes calculated from equation (3-46) are remarkably similar to the clamped-free mode shape plotted from equation (3-48) This is so even when the payload has a mass that is equal to the combined mass of the links, the shape of the first mode of vibration of the first link remains similar to the clamped-free mode shape.

Table 3.1

Dimensions of Planar Manipulator Used in Mode shape Calculations.

	Link 1	Link 2
Length	1.22 m	1.22 m
Width	0.006 m	0.006 m
Density	2.78×10^3 kg/m^3	2.78×10^3 kg/m^3
Young's Modulus	6.9×10^{10} N-m^{-2}	6.9×10^{10} N-m^{-2}

The modal frequency, however, changes dramatically as the mass of the payload changes. Table 3.2 shows the modal frequencies for all five cases. As the mass of the payload increases, the modal frequency decreases substantially.

Table 3.2 also shows modal frequencies that are calculated from the mass and stiffness matrices obtained from the assumed modes method, assuming clamped-free mode shapes.. It is seen that these values are quite close to those obtained from the finite element method.

Another comparison between the clamped-free mode shape and that calculated from the finite element method is made in Figure 3.5 for different joint configurations of the manipulator. Three different configurations are simulated and they are summarized in Table 3.3. In the first case, the manipulator links are outstretched. In the second case, they are at right angles with each other and in the third case, the second link is rotated right around so that it overlaps the first. The mode shapes are again seen to be similar to the clamped-free mode shape. Table 3.3 shows modal frequencies for all three cases when they are calculated

Table 3.2.

First Modal Frequency for Different Masses of Link 2.

Case	Mass of Link	Mass of Payload	Freq1[1]	Freq2[2]
1	0.384 Kg.	0.00 Kg.	1.88 Hz.	2.02 Hz.
2	0.384 Kg.	0.20 Kg.	1.33 Kz	1.47 Hz.
3	0.384 Kg.	0.40 Kg.	1.09 Hz.	1.21 Hz.
4	0.384 Kg.	0.60 Kg.	0.943 Hz	1.05 Hz.
5	0.384 Kg.	0.80 Kg.	0.844 Hz	0.943 Hz

[1] Freq1 = Frequency calculated by Finite Element Method.

[2] Freq2 = Frequency calculated by Assumed Modes Method.

from both the finite element and assumed modes methods. The correspondence is again quite good.

A final comparison is made between actual simulation results from both methods. Figure 3.6 shows plots of the deflection of the end point of the first link of a two-link manipulator obtained from the assumed modes and finite element methods. The manipulator is initially held vertically and the second joint is given an initial rotation of ten degrees. It is seen that the curves are almost indistinguishable from each other. These plots also verify that neglecting coupling submatrices in order to arrive at equation (3-44) is justified.

The conclusion to be drawn from these results is that the assumed modes method is to be preferred over the finite element method for simulation or control system design of flexible manipulators with uniform, high-aspect-ratio links. The analytical expressions for clamped-free mode shapes form a very good set of eigenfunctions for describing flexible link kinematics. For

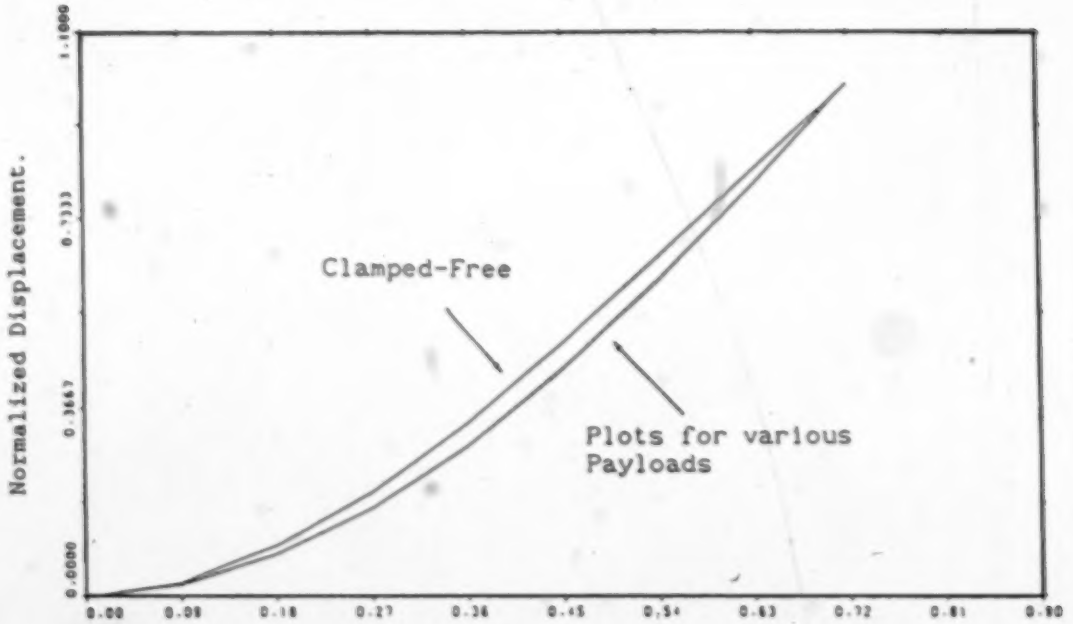


Figure 3.4 First mode shape of the first link of a two-link manipulator for varying masses of the second link.

Table 3.3.
First Modal Frequency for Different Joint Configuration

Case	q_1(Deg.)	q_2(Deg.)	Freq1[1]	Freq2[2]
1	-90.0	0.0	1.88 Hz.	2.02 Hz.
2	-90.0	+90.0	2.50 Hz.	2.66 Hz.
3	-90.0	+180.0	5.14 Hz.	5.15 Hz.

[1] Freq1 = Frequency calculated by Finite Element Method.
[2] Freq2 = Frequency calculated by Assumed Modes Method.

manipulators in which the links are not uniform, analytical expressions for clamped-free mode shapes may not be available. In these cases, the finite element method can be used to calculate the mode shapes at one configuration of the manipulator joints and this set of mode shapes used for all configurations. The assumed modes method would implicitly account for the effect of link inertia and joint configuration on modal frequencies. Model dimension when the assumed modes method is used is kept to a minimum. The algorithms of chapter 2 can then be used for quick simulation of alternative manipulator designs. Furthermore, parallel implementation of the inverse dynamics algorithm presented there would most likely make real-time computation of feedforward terms feasible. For these reasons, all simulations that are done in the rest of the thesis use the assumed modes method. The parallel algorithms presented in the next chapter are also developed with the assumed modes method in mind.

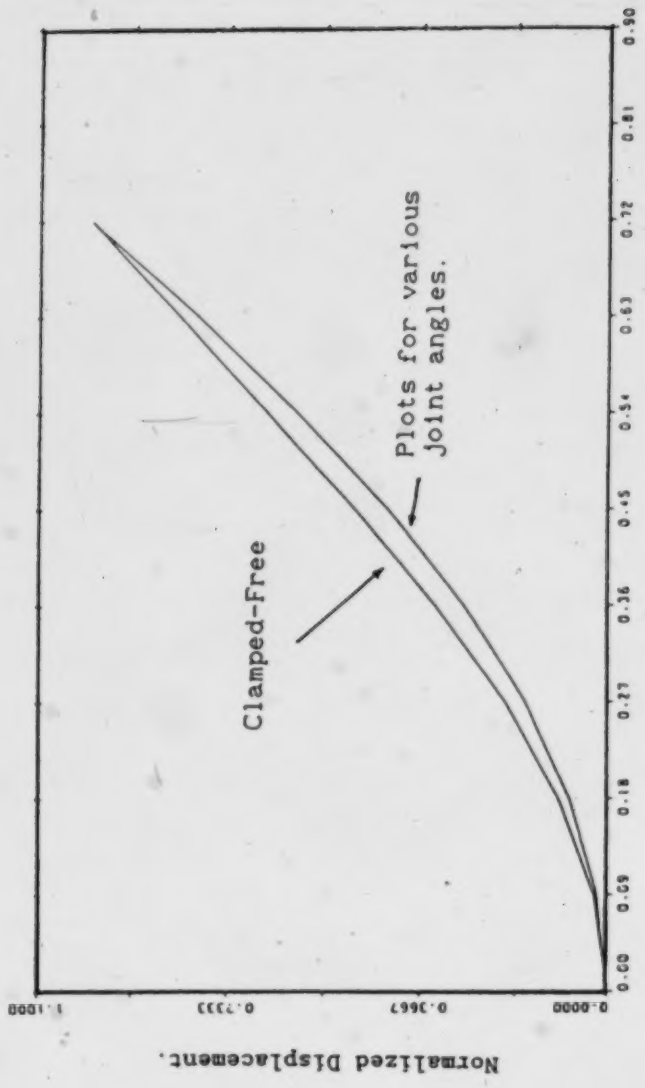


Figure 3.5. First mode shape of the first link of a two-link manipulator for several values for the second joint-angle.

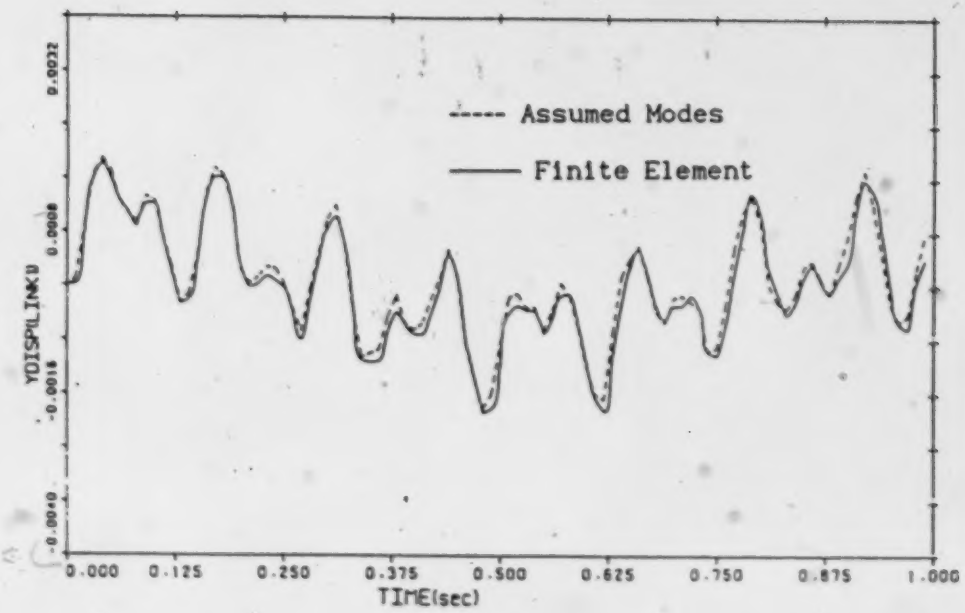


Figure 3.6. Comparison between link displacement curves obtained by the two different modeling methods.

CHAPTER 4. COMPOSITE CONTROL OF FLEXIBLE MANIPULATORS.

4.1 Introduction.

Most of the techniques for dynamic control of flexible manipulators that have been published in the open literature have been designed for, or at least tested on, a single flexible link that moves over a small distance [3,5,14]. It is not clear at this time whether these techniques would be successful in driving a multi-link flexible manipulator with reasonable accuracy over a trajectory that involves major changes in gross joint configuration. Arm dynamics vary considerably over such a trajectory. Fixed-gain controllers based on linearized dynamics at a particular point might cause unacceptable overshoots in tip position at other points along the trajectory. If a conservative design is adopted, then slow motion would occur over some parts of the trajectory. If both high speed and reasonable accuracy are desired, high-gain controllers of this type may result in system instability. Advanced control systems for flexible manipulators must therefore give a more accurate accounting of the changing dynamics.

One way of accounting for changes in dynamic parameters is to adjust the controller gains during the motion in an adaptive way. Controllers based on this method are sometimes difficult to analyse and stability is not always assured. Another method is to cancel the nonlinear dynamic effects by global feedback at each sampling instant. This is the approach taken by the popular

"computed torque" method and its variations. Their application to rigid manipulators has been studied extensively [40,75-77]. So far, computed torque control of manipulators with flexibility has been studied only in its application to arms with joint flexibility. Spong et al [91,92]. have utilized the theory of singular perturbations in order to derive modified computed torque controllers for arms with weak joint flexibility. In this chapter, the computed torque approach is extended in a general way to the design of controllers for spatial, multi-linked manipulators with flexible links.

In section 4.2, a brief description of the computed torque method for rigid links is first given. The necessary modifications to the method in order to include flexibility effects are then presented in sections 4.3 to 4.8. Simulation results to demonstrate the performance of the method are included. A multirate implementation scheme is then discussed in section 4.9.

4.2 Computed Torque Control of Rigid Manipulators.

In the classical computed torque technique for control of rigid manipulators, the command input torque is calculated as a function of desired joint accelerations $\ddot{q}_d$, velocities $\dot{q}_d$ and positions q_d, and their actual counterparts $\ddot{q}$, $\dot{q}$ and q. The system dynamics are written in the following form:

$$\underset{\sim}{\tau} = H(q)\,\ddot{q} + \underset{\sim}{h}(q, \dot{q}) \qquad (4\text{-}1)$$

where:

$\underset{\sim}{\tau}$ is the torque generated by the joint actuators,

$H(g)$ is the system inertia matrix, and

$\underset{\sim}{h}(g, \dot{g})$ is a vector or other dynamic effects.

The command control torque $\underset{\sim}{\tau}_c$ is calculated from the expression:

$$\underset{\sim}{\tau}_c = H_c(g)\left\{\ddot{g}_d + K_v\left[\dot{g}_d - \dot{g}\right] + K_p\left[\, g_d - g \,\right]\right\} + \underset{\sim}{h}_c(g, \dot{g}) \qquad (4\text{-}2)$$

where the subscript "c" indicates that the terms are calculated counterparts of the actual terms. K_v and K_p are diagonal gain matrices. It is assumed that the calculated terms are equal to the actual ones. Decoupling and linearization of the resulting error equations result, as can be seen from substituion of $\underset{\sim}{\tau}_c$ as given in equation (4-2) for $\underset{\sim}{\tau}$ in (4-1). If we define the joint error as $\underset{\sim}{e} = g_d - g$, then the error equation for each joint becomes:

$$\ddot{e}_i + K_{vi}\,\dot{e}_i + K_{pi}\,e_i = 0 \qquad (4\text{-}3)$$

The gains K_{vi} and K_{pi} can therefore be chosen to place the roots of equation (4-3) at arbitrary values. Their values are typically 20 and 100 respectively, resulting in critical dampling of the joint motion and a time constant of 0.1 seconds. Figure 4.1 is a block diagram of the control scheme.

The computed torque technique is an effective means of controlling rigid manipulator arms if the calculated dynamic terms are close in value to their actual counterparts. In effect, the open-loop system poles are cancelled by the nonlinear feedback and new system poles are determined by the choice of feedback gains. This method when used alone is not a robust approach and indeed,

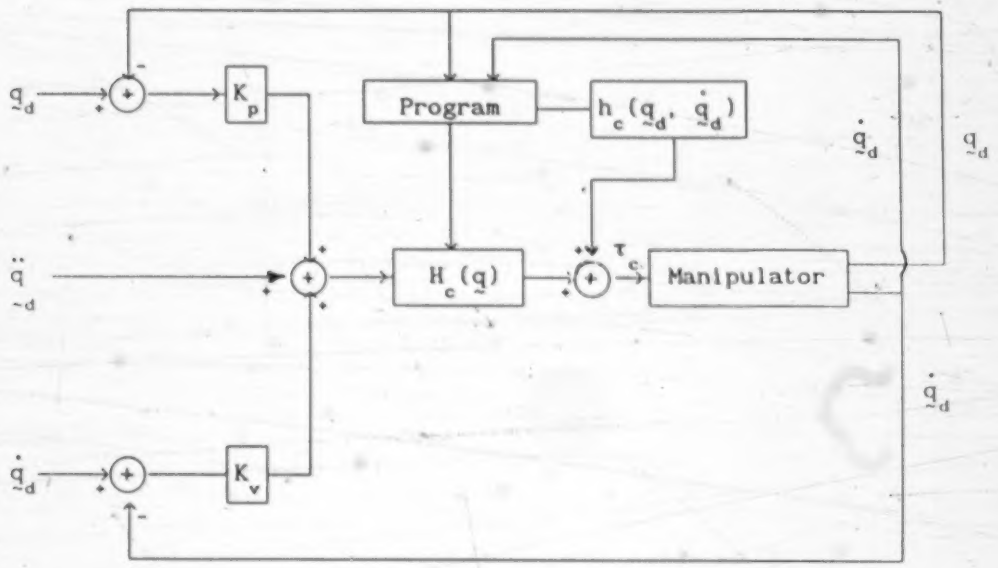


Figure 4.1 Block Diagram of the Computed Torque Controller.

if there is some variation between the calculated dynamic terms and their actual values, performance deteriorates. Several ways of improving the robustness of computed torque control have been presented in the literature [80,81]. These have added additional integral or acceleration feedback terms to the loops shown in Figure 4.1. The resulting control systems have been shown to possess good robustness properties. The overall controller is now quite attractive. The control torques can be calculated by the Newton-Euler algorithm by simply substituting the vector $\underline{\nu}(t)$ for the vector $\ddot{\underline{q}}(t)$ in the algorithm, where:

$$\underline{\nu}(t) = \ddot{\underline{q}}_d + K_v\left[\dot{\underline{q}}_d - \dot{\underline{q}}\right] + K_p\left[\underline{q}_d - \underline{q}\right] \qquad (4\text{-}4)$$

The methods for parallel computation of the Newton-Euler algorithm have now made computed torque control of rigid manipulators quite feasible.

4.3. Computed Torque Control of Flexible Manipulators.

Several researchers have designed control systems for flexible manipulators in which the desired trajectory is specified in terms of joint angles [93, 94]. The controller's task, then, is to ensure joint tracking while at the same time actively damping out link deflections. This approach is similar to the way rigid manipulators are controlled. The problems due to non-colocation of sensors and actuators are non-existent and it is relatively easy to guarantee stable motion. The major disadvantage of this method is that the two requirements of accurate joint tracking and damped

oscillations are contradictory. Rapid joint motion excites link vibrations which are only lightly damped. If the oscillations are damped by active feedback, the accuracy with which the joint follows the desired trajectory suffers. Doing both at the same time increases settling times, a situation that is generally unacceptable.

One solution to the problem lies in basing the desired trajectory on tip positions rather than joint angles. The success of this approach is dependent on the availability of cheap, fast cartesian sensors that can measure tip positions and velocities in task space coordinates. However, cartesion sensors such as vision systems tend to be quite expensive. In addition, this approach suffers from the problems that result from non-colocation of sensors and actuators. These problems are non-existent in the joint-angle tracking approach. The approach that is described in this chapter is based on an end-point trajectory but does not assume the availability of fast cartesian sensors. Instead, it assumes that individual link deflections are measureable and that a vector joining the link's base with its tip is computed. This vector has been called a "pseudolink" by Nemir in a recent paper [90]. The manipulator can be viewed as a chain of these pseudolinks. The trajectory is based on pseudolink angles rather than joint angles. The task of the controller is then to assure proper tracking of this trajectory regardless of link vibrations which can even be allowed to continue for some time after the tip has reached its final destination. Kinematic transformations back and forth along the pseudolinks allow end-point positioning without end-point sensing. The problem of non-colocation of

actuators and sensors is eliminated and simple strain gauge measurements are sufficient. Kinematic conversion between cartesian and pseudolink coordinates as in rigid manipulator control is now possible. The method is also extensible in a general way to multi-linked, spatial manipulators. This method will be referred to as the "Composite Pseudolink Controller".

In the paper by Nemir [90], a pseudolink, self tuning, adaptive controller is designed for a single link moving in a plane. The design, however, is restricted to manipulators in which only the last link is flexible. The composite pseudolink controller proposed here does not suffer from this restriction. It is based on a system decomposition into "rigid" and "flexible" subsystems. This method of decomposition is now presented.

4.4. System Decomposition.

The dynamics of a robot manipulator with flexible links can be separated into two coupled, nonlinear equations, as follows:

$$\underset{\sim}{\tau} = H_{11}(q, \underset{\sim}{\delta})\,\ddot{q} + H_{12}(q, \underset{\sim}{\delta})\,\ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_1(q, \underset{\sim}{\delta}, \dot{q}, \dot{\underset{\sim}{\delta}}) \qquad (4\text{-}5a)$$

$$\underset{\sim}{0} = H_{21}(q, \underset{\sim}{\delta})\,\ddot{q} + H_{22}(q, \underset{\sim}{\delta})\,\ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_2(q, \underset{\sim}{\delta}, \dot{q}, \dot{\underset{\sim}{\delta}}) + K\,\underset{\sim}{\delta} \qquad (4\text{-}5b)$$

where:

$\underset{\sim}{\tau} = [\tau_1, \tau_2, \cdots \tau_N]$ is the vector of joint actuator torques,

$q = [q_1, q_2, \cdots q_N]$ is the vector of joint angles,

$\underset{\sim}{\delta} = [\delta_{11}, \delta_{12}, \cdots \delta_{Nm}]$ is the vector of flexibility variables,

$\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$ are vectors of dynamics due to centrifugal, coriolis, gravity effects, etc,

$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ is the nonsingular inertia matrix, and

K is the nonsingular matrix of stiffness coefficients.

Equation (4-5a) describes the gross joint motion of the manipulator and includes the effects of link flexibility on joint motion. Equation (4-5b) describes the faster oscillations due to flexibility but also includes the effects of joint motion on flexible motion. These equations represent flexible manipulator dynamics when kinematics are represented by either the assumed modes or the finite element method. The vector $\underset{\sim}{\delta}$ is used to represent the generalized coordinates as defined in either of the methods. It should be noted that control torques are applied only at the joints and are expected to control the overall system including link flexibility.

Unlike in the rigid manipulator case, it is impossible to completely decouple all the generalized variables present in equation (4-5) from each other by nonlinear feedback. This is because only the joint variables are actuated and control of flexibility variables occurs through their coupling with the joint variables. It is possible, however, to decouple the joint variables from the flexibility variables and from each other by nonlinear feedback. Let the control torques be calculated by the following expression:

$$\underset{\sim}{\tau}_c = H_{11,c}(q, \underset{\sim}{\delta})\, \underset{\sim}{\nu}(t) + H_{12,c}(q, \underset{\sim}{\delta})\, \ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_{1,c}(q, \underset{\sim}{\delta}, \dot{q}, \dot{\underset{\sim}{\delta}}) \qquad (4\text{-}6)$$

where the subscript "c" indicates that the term is calculated and $\underset{\sim}{\nu}(t)$ is given by equation (4-4). Substituting (4-6) for $\underset{\sim}{\tau}$ in (4-5a) and assuming that the calculated terms are equal to their actual counterparts, the resulting joint equation is:

$$\ddot{\underset{\sim}{q}}(t) = \underset{\sim}{\nu}(t) \qquad (4\text{-}7)$$

The error equation is, as before, given by equation (4-3) and the controller gains can again be chosen arbitrarily.

This approach to flexible manipulator control involves measuring or estimating the flexibility variables and velocities. The assumed modes method is more attractive for use in calculating the control torques due to the lower dimension of the resulting system model. It should be noted that the vector $\ddot{\underset{\sim}{\delta}}$ is calculated from the flexibility equation (4-5b) as follows:

$$\ddot{\underset{\sim}{\delta}} = - H_{22,c}^{-1} \left[H_{21,c} \underset{\sim}{\nu}(t) + \underset{\sim}{h}_{2,c} + K_{c} \underset{\sim}{\delta} \right] \qquad (4\text{-}8)$$

The above method guarantees that the manipulator joint motion follows the commanded joint trajectory with a finite steady state error that can be made arbitrarily small. If the terms $H_{12,c}\ddot{\underset{\sim}{\delta}}$, $\underset{\sim}{h}_{1,c}$ and the contribution to the elements of $H_{11,c}$ from flexibility effects are at least bounded, then the manipulator's tip will at worst undergo non-increasing oscillatory motion when the joint angles have achieved their steady state values. This can be seen by observing the effect of the joint controller represented by equation (4-6) on the flexibility equation.

Rewrite equation (4-8) as follows:

$$\ddot{\underset{\sim}{\delta}} + A_1(\underset{\sim}{q},\underset{\sim}{\delta})\,\underset{\sim}{\delta} = T(\underset{\sim}{q},\underset{\sim}{\delta},\dot{\underset{\sim}{q}},\dot{\underset{\sim}{\delta}}) \qquad (4\text{-}9)$$

where:

$$A_1 = H_{22,c}^{-1} K_c \qquad (4\text{-}10a)$$

$$T = -H_{22,c}^{-1}\left[H_{21,c}\,\underset{\sim}{v}(t) + \underset{\sim}{h}_{2,c}\right] \qquad (4\text{-}10b)$$

It is seen that the flexible motion is driven by the coupling torques T and is initiated by $\underset{\sim}{v}(t)$ which is non-zero at the start of the motion. At steady state, however, $\underset{\sim}{v}(t) \rightarrow \underset{\sim}{0}$ and the residual excitation of the flexible system is due only to gravitational torques that are included in the vector $\underset{\sim}{h}_{2,c}$. In practice, $\underset{\sim}{h}_{2,c}$ will also include a small damping term which ensures non-increasing behaviour of the flexibility variables and, in most cases, results in decay of the oscillations.

The assumption is made here that the dependence of A_1 on the joint variables is "slower" than its dependence on the flexibility variables. Its dependence on the flexibility variables is also not very great. A_1 can therefore be considered as being "quasi-static" in the time scale of the flexibility variables. The behavior of the flexible subsystem will therefore be oscillatory since A_1 is a positive definite matrix with real eigenvalues. (Note that positive definiteness of A_1 with real eigenvalues implies marginal stability in this case, since we are dealing with a second order system. If equation (4-9) is written in the standard state-space form, the system eigenvalues will be purely imaginary).

After the decay of the oscillations, the static deflections of the links are given by the equation

$$\underset{\sim}{\delta}_{ss} = - K_c^{-1} \underset{\sim}{\bar{h}}_{2,c} \qquad (4\text{-}11)$$

where $\underset{\sim}{\bar{h}}_{2,c}$ is the vector of gravity effects only. In a control system, these static deflections must be compensated for at the end of the motion trajectory.

The description of the composite pseudolink controller design is made easier by considering a typical manipulator as we go along. We therefore digress here to describe a three-link manipulator with two flexible links. The above conclusions regarding the decoupling of the rigid subsystem from the flexible subsystem are then verified by simulation studies on this manipulator.

4.5. Example Arm.

Figure 4.2 shows the shape of the three-link, spatial manipulator that is simulated. It has a kinematic configuration that is similar to the PUMA 560 shown in Figure 1.1. The first link is mounted vertically and rotates about its base. This link is considered to be rigid for the purpose of the simulation. The other two links have high aspect ratios and are quite flexible. Transverse and torsional vibrations are included in the simulation. The manipulator parameters are summarized in Table 4.1. The payload is represented by a point mass of 0.3 kilogram, giving a mass to payload ratio of about 8.1 to 1, compared with 50 to 1 for the PUMA 560.

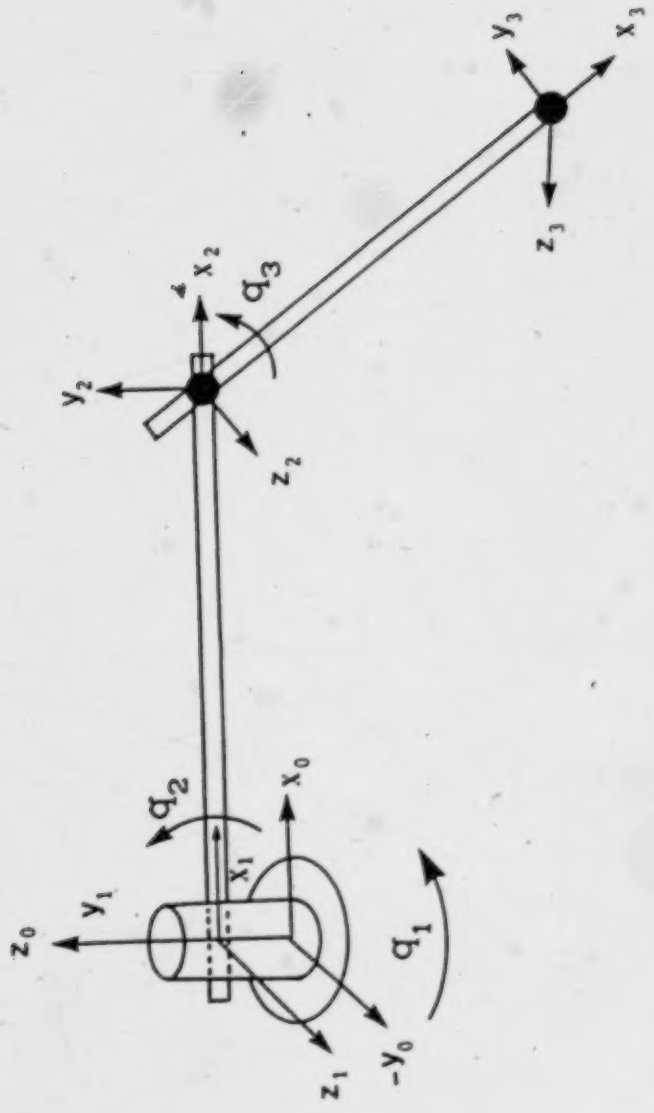


Figure 4.2. Three-link manipulator used for the simulations.

Table 4.1. Link Parameters of Example Arm.

	Link 1	Link 2	Link 3
mass	0.79 kg	1.07 kg.	0.69 kg
length	0.10 m.	1.22 m.	1.22 m.
radius	0.03 m.	0.010 m	0.008 m
twist	90 deg.	0 deg.	0 deg.
offset	0.10 m.	0.0 m.	0.0 m.
f_1^*	-	3.75 Hz.	1.94 Hz

* f_1 is the fundamental frequency of vibration in the y-direction when payload mass is 0.3kg.

4.6. Simulation Results With Joint Controller Only.

In this simulation run, only the joint controller given by equation (4-6) is applied. Figure 4.3(a-c) show the responses of the manipulator joints to commanded steps of 10 degrees for each joint. The controller gains are: K_{pi}= 100, K_{vi}= 20 for all three joints. These gains correspond to critically damped motion with closed-loop poles at (-10, 0).

It is clear that flexibility effects are completely decoupled from these responses and that the joint angles follow the desired trajectories quite well.

Figure 4.4(a-d) show the tip deflections of the two flexible links in the y and z directions. From these plots it is seen that after the joints have reached their steady states, the tip motion of the links continues to be oscillatory. The oscillations are sustained since structural damping is absent from the model used

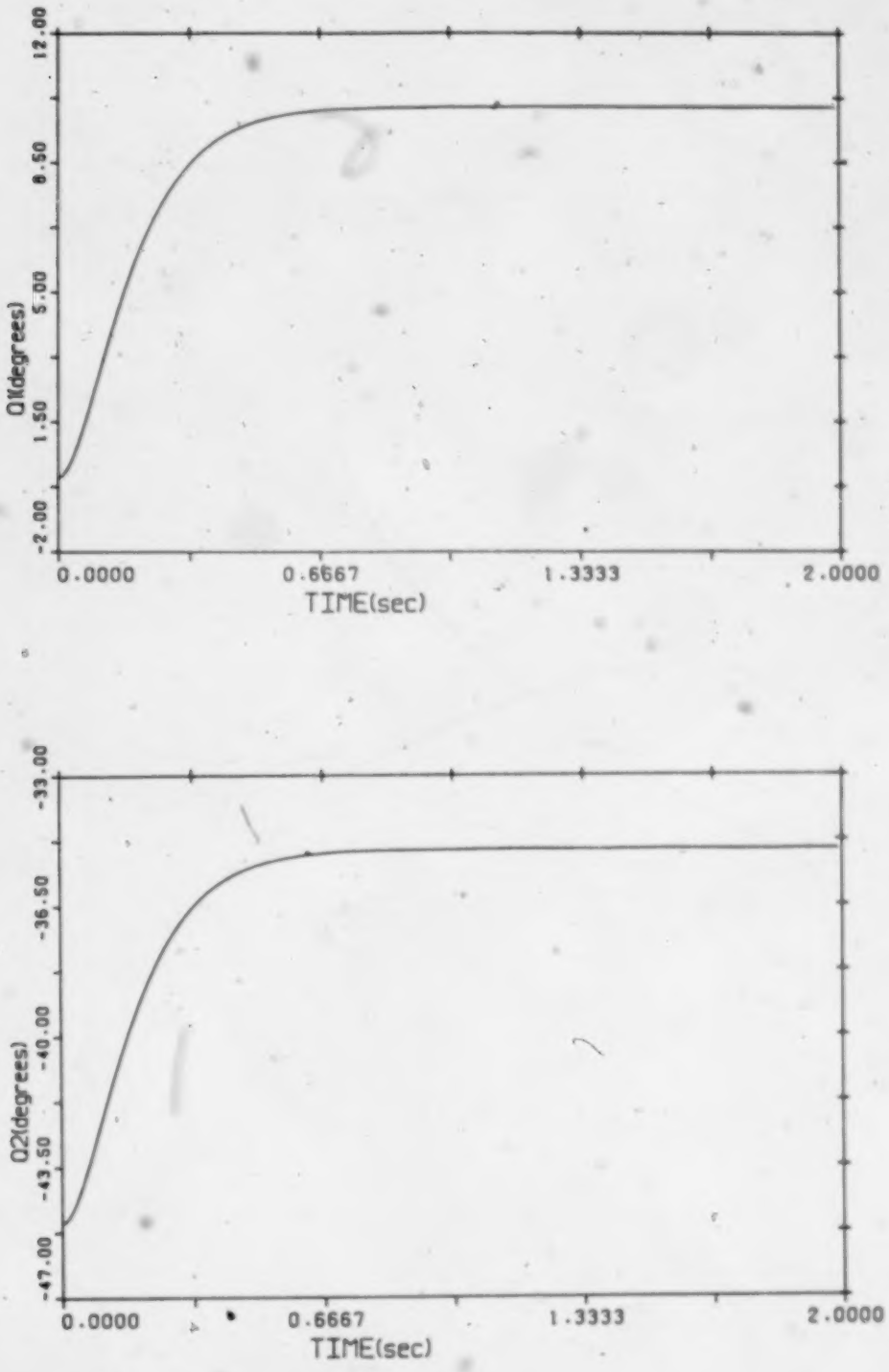


Figure 4.3. Joint angle responses when the decoupling signal only is used for control.

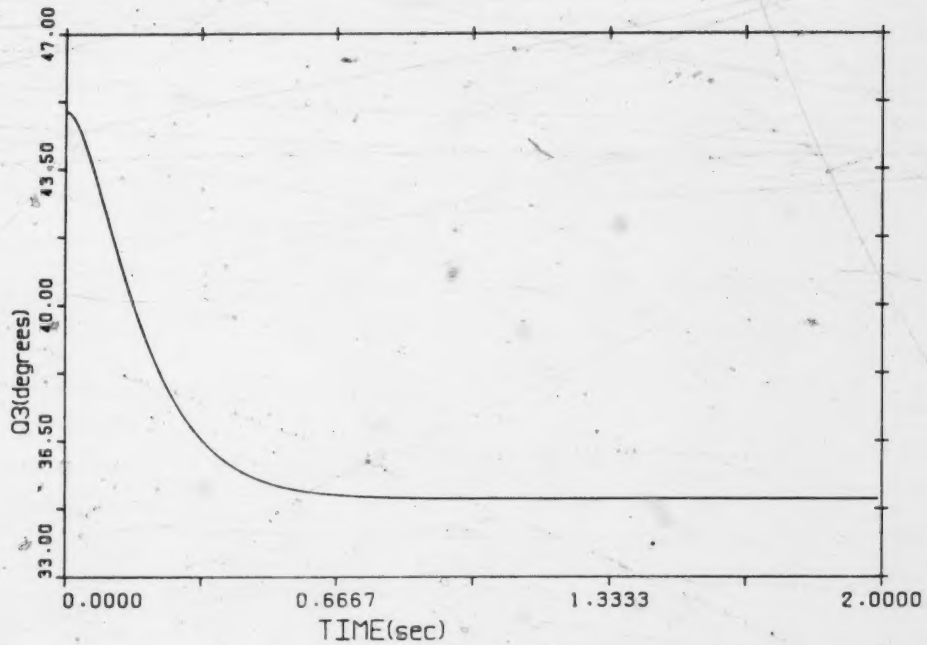


Figure 4.3. (Cont'd)

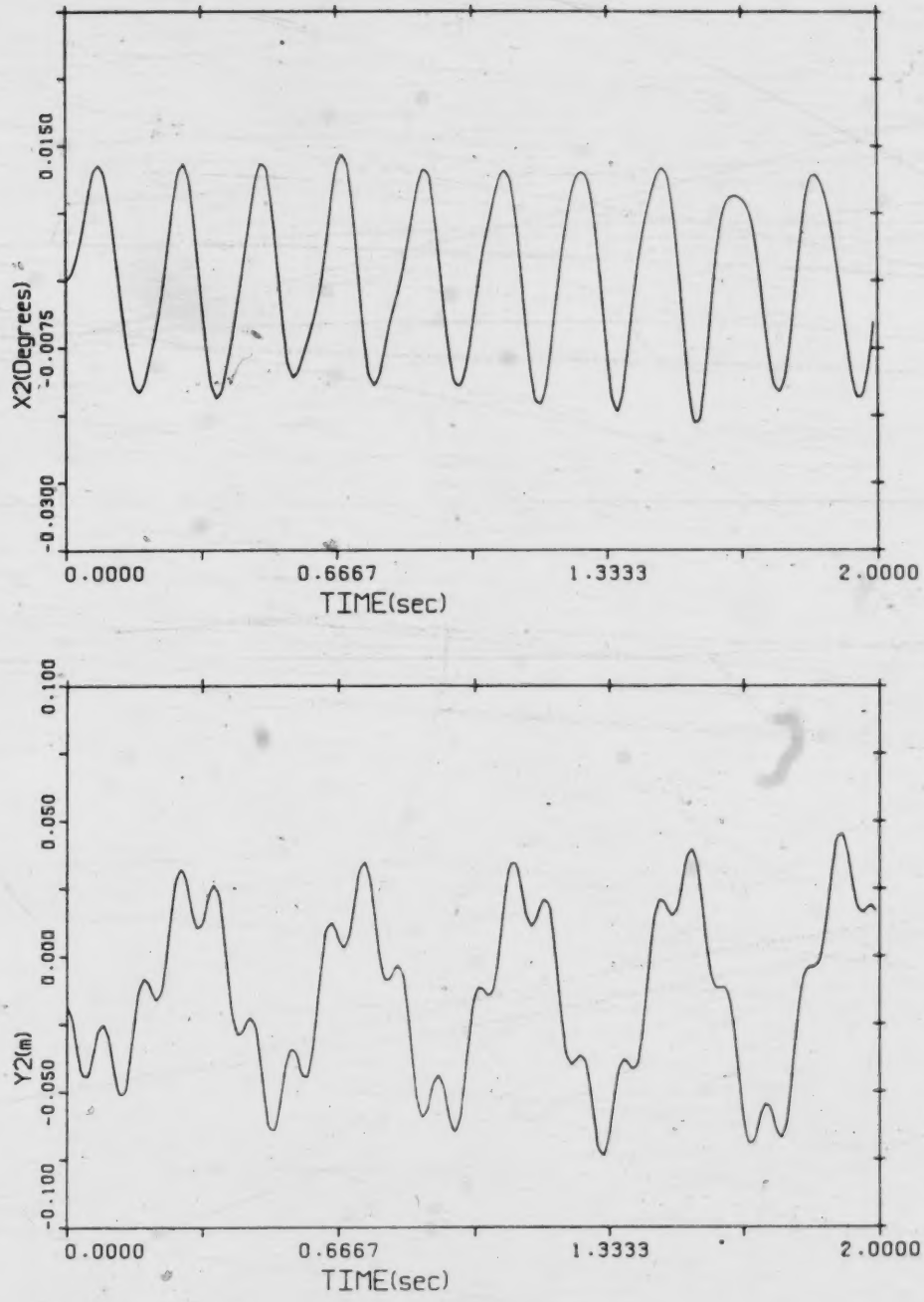
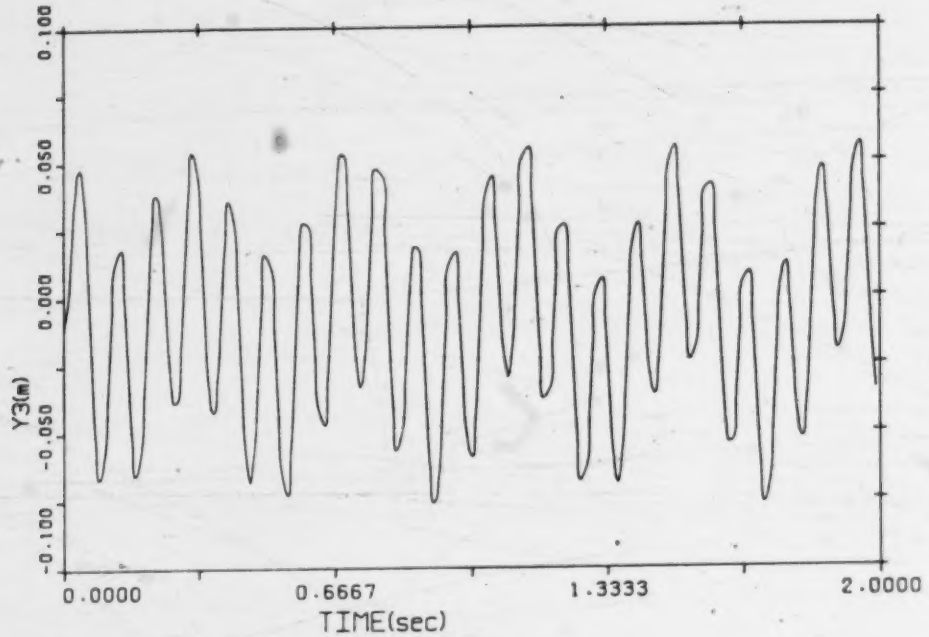


Figure 4.4. Tip deflections of the two flexible links of the three-link manipulator under decoupling control only. (Y,Z = displacements in the y and z directions, and X = torsional angle.)



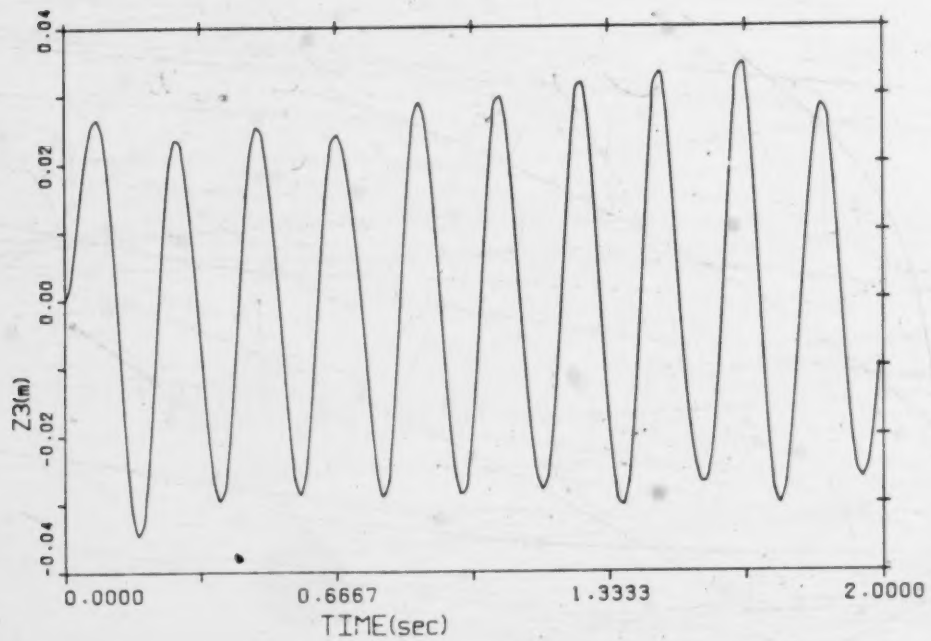


Figure 4.4. (Cont'd)

for simulation. The results of the simulation confirm the validity of the model decomposition discussed earlier. This lays the foundation for the composite control scheme whose objective is to ensure that the pseudolink angles follow the commanded trajectory with little or no error.

4.7. Design of Pseudolink Controller.

If the composite control method proposed by Spong et al [91,92] for controlling flexible joints is followed at this point, then the next step in the design procedure would be to synthesize a flexible motion controller that damps out the flexible oscillations observed in Figure 4.4. This can be done by writing $\underset{\sim}{\nu}(t)$ as follows:

$$\underset{\sim}{\nu}(t) = \underset{\sim}{\nu}_r(t) + \underset{\sim}{\nu}_f(t) \qquad (4\text{-}14)$$

where $\underset{\sim}{\nu}_r(t)$ is identical to $\underset{\sim}{\nu}(t)$ as given by equation (4-4) and is designed to drive the joints along their desired trajectory. $\underset{\sim}{\nu}_f(t)$ must be designed to actively damp out flexible vibrations while not throwing the joints too far off their desired trajectories. Substitution of equation (4-14) into equation (4-6) results in the closed-loop joint equation:

$$\ddot{\underset{\sim}{q}}(t) = \underset{\sim}{\nu}_r(t) + \underset{\sim}{\nu}_f(t) \qquad (4\text{-}15)$$

The flexibility equation becomes:

$$\ddot{\underset{\sim}{\delta}}_f + A_1(\underset{\sim}{q},\underset{\sim}{\delta})\,\underset{\sim}{\delta}_f + B_1(\underset{\sim}{q},\underset{\sim}{\delta})\,\underset{\sim}{\nu}_f(t) = \underset{\sim}{0} \qquad (4\text{-}16)$$

where:

$$B_1 = H_{22,c}^{-1} H_{21,c} \tag{4-17a}$$

$$A_1 = H_{22,c}^{-1} K_c \tag{4-17b}$$

$$\underset{\sim}{\delta}_f = \underset{\sim}{\delta} - \bar{\underset{\sim}{\delta}} \tag{4-17c}$$

$$\bar{\underset{\sim}{\delta}} = - H_{21,c} \underset{\sim}{\nu}_r(t) - \underset{\sim}{h}_{2,c} \tag{4-17d}$$

The benefit of writing the flexible subsystem in the form given in equation (4-16) is that the equation becomes "quasi-linear" linear in the flexibility variables. The matrix coefficients are still nonlinearly dependent on both the joint and flexibility variables, more so on the joint than on the flexibility variables. For the case where the joint variables vary at a slower rate than the flexibility variables, these coefficients can be considered as being "quasi-constant". Equation (4-16) then approximately represents a linear, time-invariant set of differential equations that is both controllable and observable. The control vector $\underset{\sim}{\nu}_f(t)$ can therefore be calculated using one of the many schemes available for designing control systems for linear systems.

The poles of the closed-loop flexible subsystem can be chosen such that $\underset{\sim}{\nu}_f(t) \rightarrow \underset{\sim}{0}$ quite rapidly. In the traditional singular perturbation approach [91,92], $\bar{\underset{\sim}{\delta}}$ is given by a zero'th order approximation to equation (4-16). That is, the expression on the right hand side of (4-16) is written as an asymptotic expansion in terms of some small parameter (μ) and only the zero'th order term is retained. This term is dependent on the joint angles and not on the flexibility variables. A controller that forces the variables

$\underset{\sim}{\delta}(t)$ to follow the quasi-steady state $\bar{\underset{\sim}{\delta}}(t)$ as defined in this manner damps the flexible oscillations out. Except for a small steady state error in manipulator tip position due to static deflection, the manipulator is forced to behave as though it were rigid for most of its motion.

The major disadvantage of this approach is that for success, the parameter μ must be quite small. For links with significant flexibility, μ may be too large and the zero'th order approximation to $\bar{\underset{\sim}{\delta}}$ may not be sufficiently accurate. In this case, the flexible oscillations may not decay fast enough and the manipulator tip may still be vibrating at the end of the trajectory.

The solution to this problem is to rewrite equations (4-6) and (4-16) in terms of pseudolinks. For the three-link manipulator shown in Figure 4.2, we consider the flexible deflections of the second and third links in the transverse y-direction to be controlled by the second and third joint actuators respectively. For these links, the pseudolink angles in the y directions (γ_{2y} and γ_{3y}) are given by the following expressions:

$$\gamma_{2y} = q_2 + \beta_{2y} \qquad (4\text{-}19)$$

$$\gamma_{3y} = q_3 + \beta_{3y} + \phi_{2y} - \beta_{2y} \qquad (4\text{-}20)$$

where all angles are defined in Figure 4.5. The angles β_{2y} and β_{3y} are approximated by the following expressions:

$$\beta_{2y} = \frac{\Delta_{2y}}{\ell_2}; \quad \beta_{3y} = \frac{\Delta_{3y}}{\ell_3}; \qquad (4\text{-}21)$$

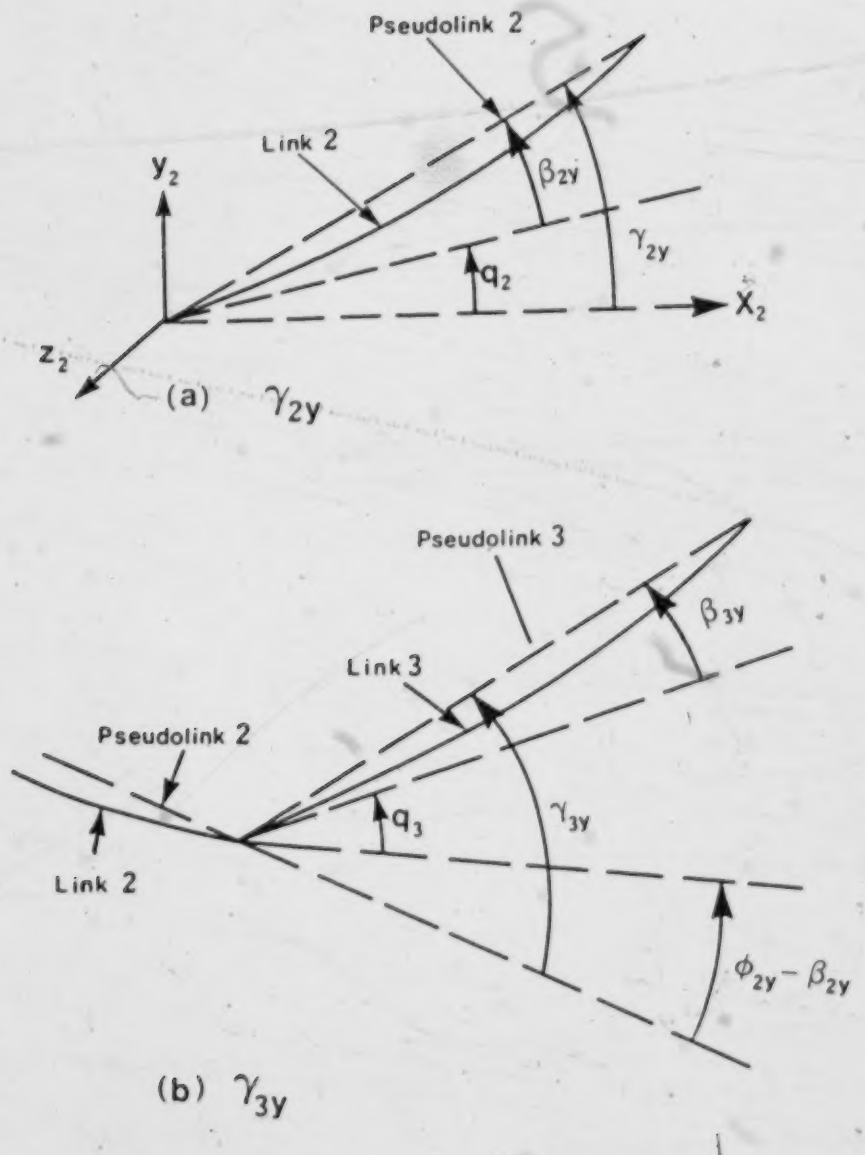
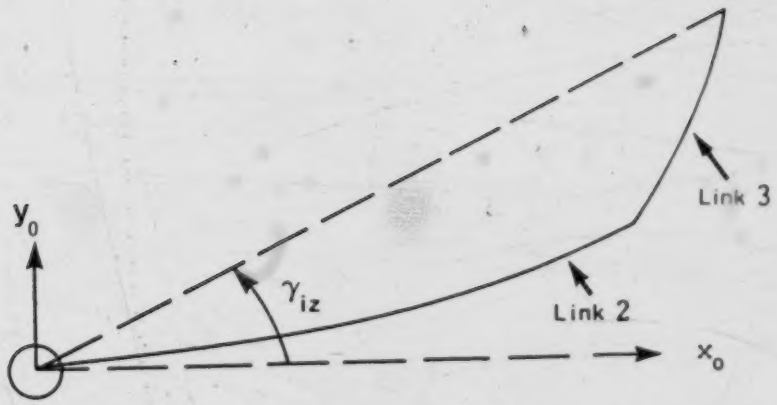


Figure 4.5. Definition of the Pseudolink angles.



(c) γ_{iz}

Figure 4.5 (cont'd)

where again all terms are defined in Figure 4.5. The pseudolink angles are therefore given by the sum of the joint angles and simple linear combinations of the flexibility variables. This makes modification of the manipulator dynamic equations quite easy and facilitates the pseudolink controller design.

The only means of controlling flexible deflections in the x-z plane, and torsion of the second link, is through the actuator of the first joint. The pseudolink angle γ_{1z} is shown in Figure 4.5 as well. It is seen that kinematic transformations from the second link down to the base are necessary in order to calculate γ_{1z}. Fortunately, some simplifications can be made after the kinematic transformations that allow γ_{1z} to be written as the sum of the joint angle (q_1) and a linear combination of the flexibility variables. Second and higher powers of the deflection variables are deleted during the transformations. Also, coupling terms between deflections in the x-y and x-z planes are neglected. The resulting expression becomes:

$$\gamma_{1z} = q_1 + \frac{1}{P}\left[-\Delta_{2z} + \phi_{2y}C_3\ell_3 - \phi_{2x}S_3\ell_3 + \Delta_{3z}\right] \quad (4\text{-}22)$$

where:

$P = C_2\ell_2 + C_3\ell_3$

$C_1 = \cos(q_1)$

$S_1 = \sin(q_1)$

ℓ_2, ℓ_3 = lengths of links 2 and 3 respectively,

Δ_{1y}, Δ_{1z} = deflections of the tip of the $1'^{th}$ link in the y and z directions respectively,

ϕ_{1y}, ϕ_{1z} = angular deflections of the tip of the $1'^{th}$ link

about the y and z axes respectively,

ϕ_{2x} = torsional angle of the second link. Note that the third link does not undergo torsional rotation for this manipulator.

q_i = i^{th} joint angle.

At this point, advantage is taken of the fact that the joint angles vary at a slower rate than the flexibility variables. The value of P can therefore be considered to be quasi-constant, and recalculated a small number of times over a trajectory.

Combining equations (4-19), (4-20) and (4-22), we obtain:

$$\underset{\sim}{\gamma}(t) = \underset{\sim}{q}(t) + F\,\underset{\sim}{\delta}(t) \qquad (4\text{-}23)$$

where F is a piecewise constant matrix, and:

$$\underset{\sim}{\gamma} = \left[\gamma_{1x}, \gamma_{2y}, \gamma_{3y}\right]^T$$

$$\underset{\sim}{q} = \left[q_1, q_2, q_3\right]^T$$

Equation (4-23) in now substituted into (4-5), resulting in:

$$\underset{\sim}{\tau} = H_{11}\,\ddot{\underset{\sim}{\gamma}} + \left[H_{12} - H_{11}F\right]\ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_1 \qquad (4\text{-}24a)$$

$$\underset{\sim}{0} = H_{21}\,\ddot{\underset{\sim}{\gamma}} + \left[H_{22} - H_{21}F\right]\ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_2 + K\,\underset{\sim}{\delta} \qquad (4\text{-}24b)$$

The joint controller then becomes:

$$\underset{\sim}{\tau}_c = H_{11,c}\,\underset{\sim}{\nu}_p(t) + \left[H_{12,c} - H_{11,c}\,F\right]\ddot{\underset{\sim}{\delta}} + \underset{\sim}{h}_{1,c} \qquad (4\text{-}25)$$

where:

$$\underset{\sim}{\nu}_p(t) = \ddot{\underset{\sim}{\gamma}}_d + K_v\left[\dot{\underset{\sim}{\gamma}}_d - \dot{\underset{\sim}{\gamma}}\right] + K_p\left[\underset{\sim}{\gamma}_d - \underset{\sim}{\gamma}\right] \qquad (4\text{-}26)$$

Substituting (4-25) and (4-26) into (4-24a) results in the closed-loop error equation becoming:

$$\ddot{\underset{\sim}{e}} + K_v\,\dot{\underset{\sim}{e}} + K_p\,\underset{\sim}{e} = \underset{\sim}{0} \qquad (4\text{-}27)$$

where the trajectory error is now defined as $\underset{\sim}{e} = \underset{\sim}{\gamma}_d - \underset{\sim}{\gamma}$. The gains K_p and K_v can again be chosen to place the closed-loop poles of the joint subsystem at arbitrary locations.

The joint angles would follow the desired trajectory under the control given by equation (4-25), provided the resulting flexible subsystem is at least marginally stable. However, this is not so in the majority of cases. The flexible subsystem is given by the following equation:

$$\ddot{\underset{\sim}{\delta}}_f(t) + A_2\,\underset{\sim}{\delta}_f(t) = \underset{\sim}{0} \qquad (4\text{-}28)$$

where:

$$\underset{\sim}{\delta}_f = \underset{\sim}{\delta} - \bar{\underset{\sim}{\delta}} \qquad (4\text{-}29a)$$

$$\bar{\underset{\sim}{\delta}}_f = -\left[H_{21}\,\underset{\sim}{\nu}_p(t) + \underset{\sim}{h}_{2,c}\right] \qquad (4\text{-}29b)$$

$$A_2 = \left[H_{22} - H_{21}F\right]^{-1} K \qquad (4\text{-}29c)$$

The frequencies of vibration of the flexible links are assumed to be sufficiently high such that the matrix A_2 can be considered "quasi-static" in the time scale of the flexibility variables. For

the three-link manipulator shown in Figure 4.2, some of the eigenvalues of A_2 turn out to be negative. This means that some of the poles of the system are in the right half-plane, indicating that the system is unstable. (Note that we are again dealing here with a second order differential equation that is not written in standard state-space form. If the equation is converted into standard form, represented by a first order differential equation, some of the eigenvalues of the system matrix would have positive real parts.) The system, however, is controllable at all points except a finite number of singular points. These singular points would normally be avoided during the trajectory planning stage anyway. A composite controller that stabilizes the flexible subsystem can therefore be synthesized by writing $\underset{\sim}{\nu}_p(t)$ as follows:

$$\underset{\sim}{\nu}_p(t) = \underset{\sim}{\nu}_{p,r}(t) + \underset{\sim}{\nu}_{p,f}(t) \qquad (4\text{-}30)$$

where $\underset{\sim}{\nu}_{p,r}(t)$ is the same as $\underset{\sim}{\nu}_p(t)$ in (4-26). The closed-loop joint equation becomes:

$$\ddot{\underset{\sim}{\gamma}}(t) = \underset{\sim}{\nu}_{p,r}(t) + \underset{\sim}{\nu}_{p,f}(t) \qquad (4\text{-}31)$$

and the open-loop flexible subsystem becomes

$$\ddot{\underset{\sim}{\delta}}_f(t) + A_2 \underset{\sim}{\delta}_f(t) + B_2 \underset{\sim}{\nu}_{p,f}(t) = \underset{\sim}{0} \qquad (4\text{-}32)$$

where:

$$B_2 = \left[H_{22} - H_{21} F\right]^{-1} H_{21} \qquad (4\text{-}33)$$

The flexible subsystem can now be controlled by state feedback using one of the many techniques avaflable for synthesizing the feedback gains of a linear system. Normally, the gains will be chosen such that $\underset{\sim}{\delta}_f \rightarrow \underset{\sim}{0}$ quite rapidly. This does not necessarily mean that the oscillations due to flexibility decay as rapidly. It means that the quasi-steady state is tracked with small rise times and settling times. After this is accomplished, the pseudolink angles would track the desired trajectory with the desired accuracy.

The major advantage of this method is that the quasi-steady state $\underset{\sim}{\bar{\delta}}(t)$ need not be approximated by the zero'th order term in an asymtotic expansion in terms of a small parameter. In fact, no mention at all of a small parameter is necessary. The value of $\underset{\sim}{\bar{\delta}}(t)$ can be calculated directly from equation (4-29b). The pseudolinks would track the desired trajectory even if the value of $\underset{\sim}{\bar{\delta}}(t)$ is non-zero at the end of the trajectory. The manipulator joint may still be oscillating at this point but the tip would be at its desired position.

4.8. Simulation Results.

The assumed modes method described in chapter 2 is used to model the three-link manipulator shown in Figure 4.2. Link flexibility is represented by the first clamped-free mode shape. The open-loop poles of the flexible subsystem are given in Table 4.2. Note that some of them have positive real parts as asserted earlier. An optimal method [97] is used to determine the feedback gains of the flexible subsystem. The closed-loop poles are given

Table 4.2

Open-loop poles of the flexible subsystem.

Y-Direction		Z-Direction		Torsion	
Real	Imag.	Real	imag.	Real	Imag.
0.0	234.5	0.0	47.14	0.0	54.8
0.0	-234.5	0.0	-47.14	0.0	-54.8
31.5	0.0	27.1	0.0	0.0	65.5
-31.5	0.0	-27.1	0.0	0.0	-65.5

Table 4.3

Closed-loop poles of the flexible subsystem under Pseudolink control

Y-Direction		Z-Direction		Torsion	
Real	Imag.	Real	imag.	Real	Imag.
-780.2	0.0	-27.9	0.0	0.0	54.8
-49.1	7.9	-27.7	0.0	0.0	-54.8
-49.1	-7.9	-0.03	39.1	-0.016	65.5
-28.0	0.0	-0.03	-39.1	-0.016	-65.5

in Table 4.3. A payload of 0.3kg. is assumed to be attached to the manipulator's tip.

The desired motion profile of each joint is as represented by the curve shown in Figure 4.6. The maximum angular displacement is the same for each joint and is 90 degrees, but the initial joint angles are all different. The initial joint angles are: $q_1 = 0$ degrees, $q_2 = -45$ degrees, and $q_3 = 45$ degrees. The curve in Figure 4.6 is obtained by specifying a constant acceleration of 90 deg.s^{-2} for the first second, then suddenly reversing its direction for the latter second of the trajectory. The initial velocities are all set to zero and the initial desired joint angles are set equal to the initial angles. The desired angles (q_{id}) and velocities ($\dot{q}_{id}$) at each integration time step is calculated by the following formulas:

$$q_{id}(t+dt) = q_{id}(t) + \dot{q}_{id}(t)\, dt \qquad (4\text{-}12)$$

$$\dot{q}_{id}(t+dt) = \dot{q}_{id}(t) + a_i(t)\, dt \qquad (4\text{-}13)$$

where a_i is the desired acceleration. The desired acceleration of the third joint is -90 deg.s^{-2} and, as mentioned above, is 90 deg.s^{-2} for each of the first and second joints. The controller gains are: $K_{pi} = 100$, $K_{vi} = 20$. These are typical gains used in controlling rigid manipulators and correspond to a critically damped response. The closed-loop poles of the system are at (-10,0) in the s-plane.

The desired trajectory of each pseudolink angle is the same as the desired joint angle described in section 4.6. The initial

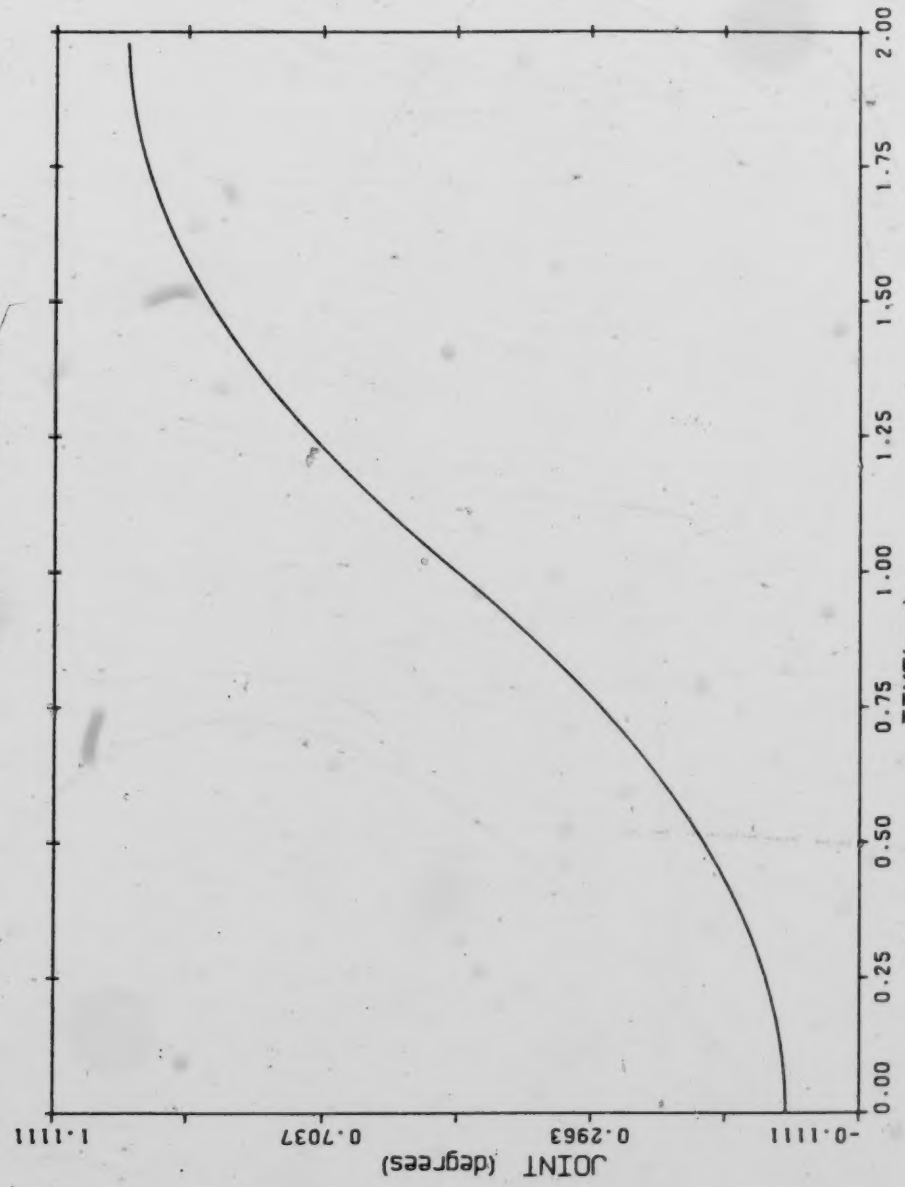


Figure 4.6. Joint-angle trajectory prototype for all three joints. (1.0 on the vertical scale corresponds to maximum joint displacement)

velocities are all zero for both joint and flexibility. The initial flexibility variables are set to values that approximate the static deflections of the links at the starting position. The initial joint angles are: $q_1 = 0$, $q_2 = -45$ degrees, and $q_3 = 45$ degrees.

In order to assess the performance of the composite pseudolink controller, the motion of a rigid manipulator that is controlled by the traditional computed torque method is simulated. The kinematic configuration of this manipulator is the same as that of the flexible manipulator. The radii of the second and third links, however, are increased to 0.05 meters each. This would be necessary in order to stiffen the links sufficiently such that they can be considered rigid.

Figure 4.7 shows the actual trajectories of the pseudolink angles in the case of the flexible manipulator, and the joint angles in the case of the rigid manipulator. It is seen that, as predicted, apart from initial transients, the pseudolink angles track the joint angle trajectories of the rigid manipulator quite well. Manipulator speed is manitained without suffering too much by way of reduced tracking accuracy.

The main advantage of lightweight arms is the ability to maintain manipulator speed and stability with low power actuators. This advantage is demonstrated in Figure 4.8. In this graph, the actuator torques of each of the three joints for both the flexible manipulator and the rigid one are plotted. The torque profiles for the other actuators are similar. It is seen that at least a five-fold reduction in actuator torque can be realized by reducing the radii of the links.

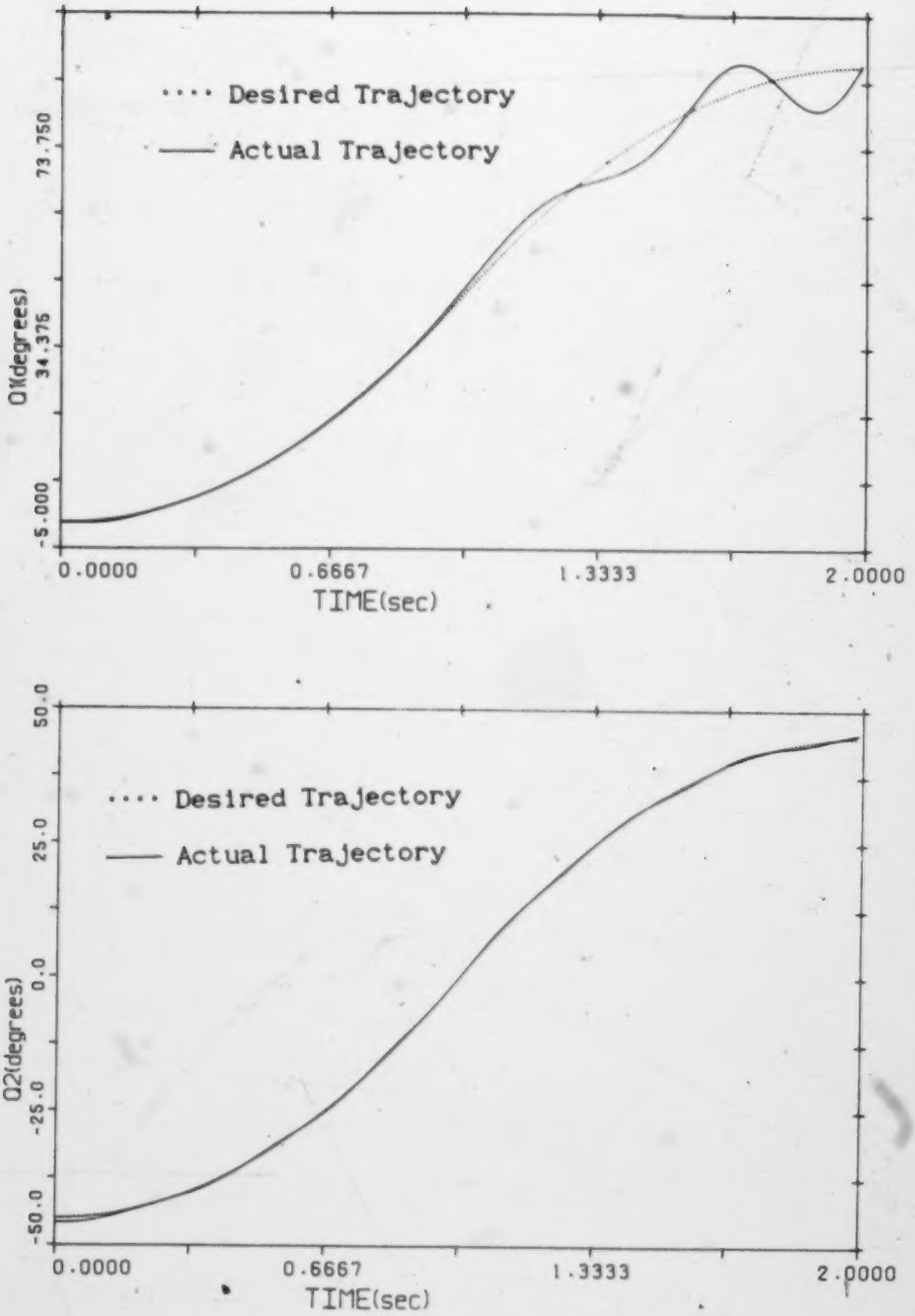


Figure 4.7 Simulation results showing the Pseudolink-angle trajectory of the second link, compared with the desired trajectory.

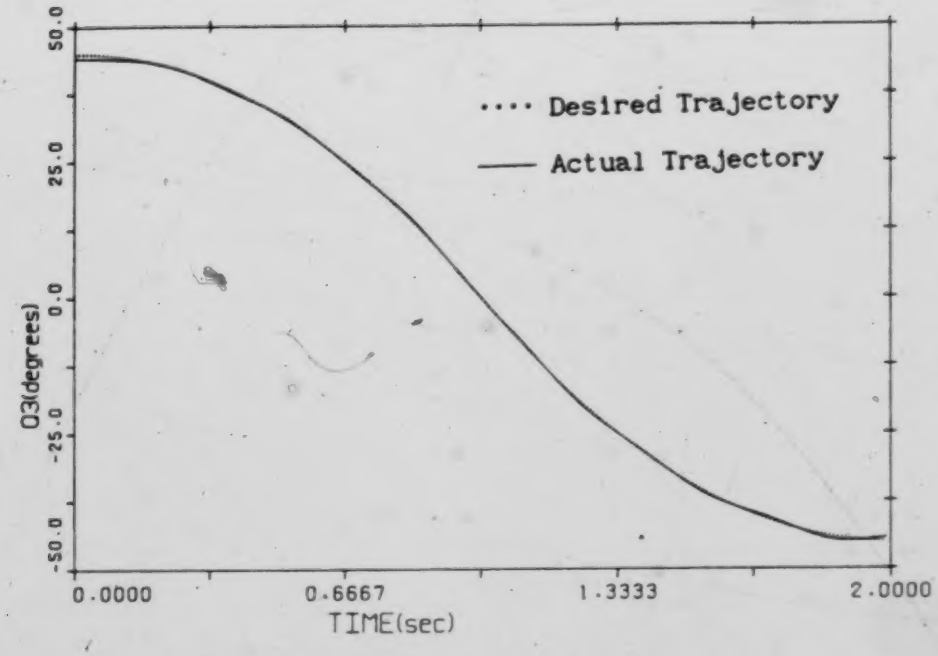


Figure 4.7. (Cont'd)

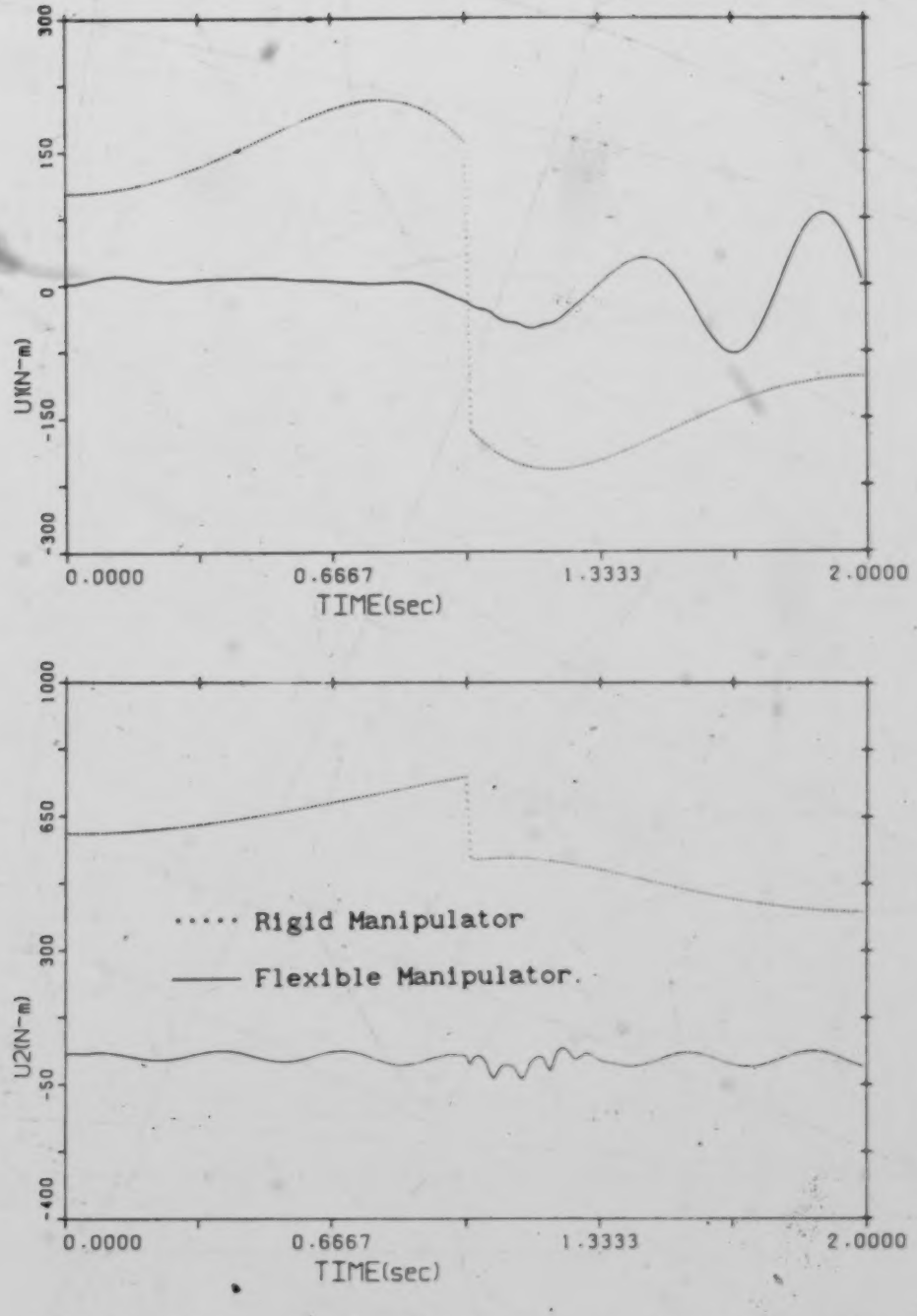


Figure 4.8. Simulation results showing the required torques for a flexible manipulator under pseudolink control, compared with the required torques for rigid manipulator control.

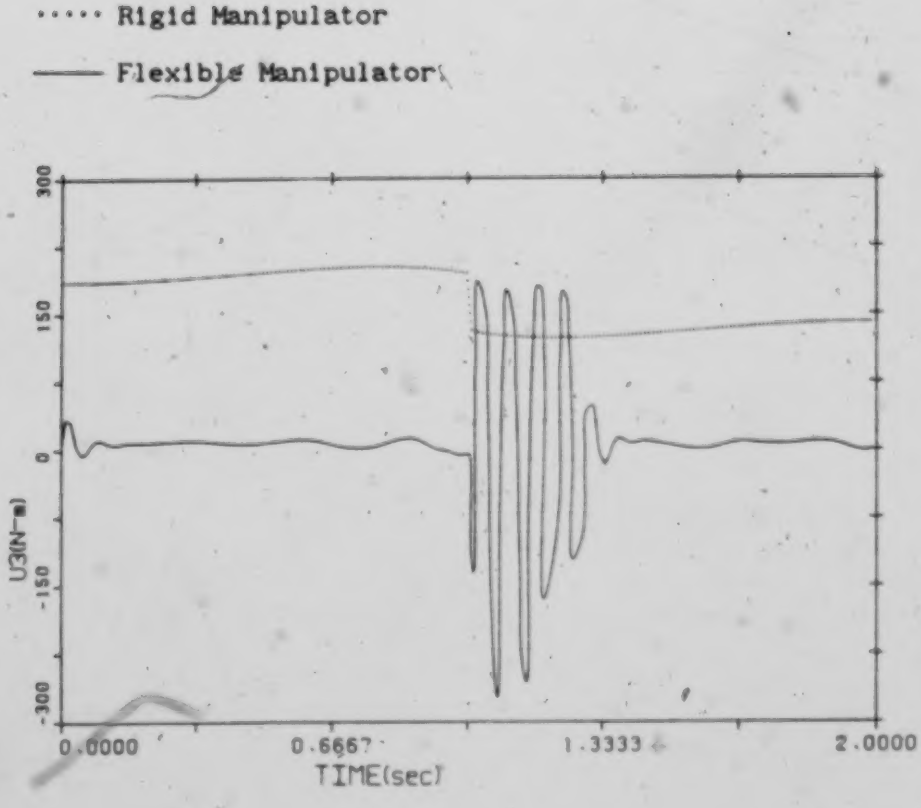


Figure 4.8. (Cont'd)

4.9. Implementational Considerations.

Implementation of the composite pseudolink controller requires real-time calculation of the inverse dynamics using a dynamic model that includes flexibility. Implementation of the Newton-Euler-like, inverse algorithm that was presented in chapter 2 on a uniprocessor may not be fast enough for real-time control. Ways of either reducing the computational requirements of a computed torque controller, or of speeding up the calculations, are needed. Techniques of parallel processing the calculations help to speed them up. In this section, we describe a novel implementation method that reduces the computational requirements of computed torque methods for flexible manipulators. This method was first presented at the 1986 International Conference on Systems, Man and Cybernetics [96].

As was observed earlier, flexible dynamics give rise to time constants that are generally smaller than the time constants associated with the gross motion of the joints. If the links are not too flexible, the singular perturbation approach allows separation of the overall model into two subsystems, a slow (rigid) and a fast (flexible) subsystem in which the slow subsystem is identical to the manipulator dynamical system if the links were rigid. The dynamical matrices of the fast subsystem are dependent on the slow variables only, and can be considered as constant in the fast time scale. The slow subsystem can be controlled at a slow rate, and the fast subsystem at a fast rate. This has the advantage that the inverse dynamics calculation, which requires the longest CPU time, can now be given time required. The fast controller merely requires the multiplication

of gain matrices by the flexibility variables and velocities and this can be done quite easily at a fast rate. The overall implication is that computational requirements for controller implementation can be considerably reduced.

4.9.1. Sampling Rate Considerations.

Sampling rate considerations are very important in controller implementation using digital computers. Rates of 50Hz. to 60Hz. have been suggested for control of rigid manipulators. Much higher rates than these would be necessary for control of flexible manipulators. It has been demonstrated in [96] that if all considerations of joint or link flexibility were omitted from the dynamical equations, then a sampling rate of 10 Hertz would still guarantee stability of the (rigid) system when a computed torque controller is used with the same gains as used in the simulation in the previous section. This rate is considerably less than the rates mentioned above. Rates of 50 Hertz were necessary only because the joints of the manipulators showed oscillatory characteristics and the sampling rate had to be chosen to be five to ten times the frequency of these oscillations. If joint and link flexibility were grouped together in the fast subsystem that resulted from the system decomposition that used the singular perturbation ideas, then the composite controller can be implemented with two sampling rates in the hardware. In the worst-case scenario, the sampling rate used to control the rigid subsystem can be maintained at 50 Hertz, while that part of the hardware that controls flexibility can operate at the higher sampling rate. This multirate control implementation is depicted

in the block diagram shown in Figure 4.9.

The feasibility of this scheme is demonstrated by simulation of a simple, one-link manipulator arm with a concentrated load at its tip. The link is shown in Figure 4.10. It has a length of one meter and a mass of one kilogram. The payload mass is two kilograms. The flexibility parameters are chosen such that the maximum deflection of the link over a step of one radian is about five percent of the link's length. The open-loop natural frequency of the fast subsystem is about 10 Hertz. The assumed modes method with one flexible mode is used to simulate the link. The slow subsystem is controlled by a computed torque method with controller gains as follows: $K_p = 100$, $K_v = 20$. Control is based on driving the joints to follow a step in joint position and at the same time damping out the link's oscillations. A small damping factor of three percent is introduced into the dynamics of the flexible subsystem.

In the first test shown in Figure 4.11, the fast controller is omitted and the slow (computed torque) controller is run at the nominal rate of 50 hertz. The system is clearly stable, with the joint angle following the commanded input as desired. The oscillations of the link decay as a result of the damping introduced as part of the dynamic model. However, it was found that as the sampling rate was reduced to about 25 Hertz, the system lost stability. This is clearly a result of the effect of flexibility on the system dynamics. This effect was not taken into consideration by the control system. If this effect were not present, sampling rates down to 10 Hertz would not have resulted in system instability. Furthermore, the damping that was

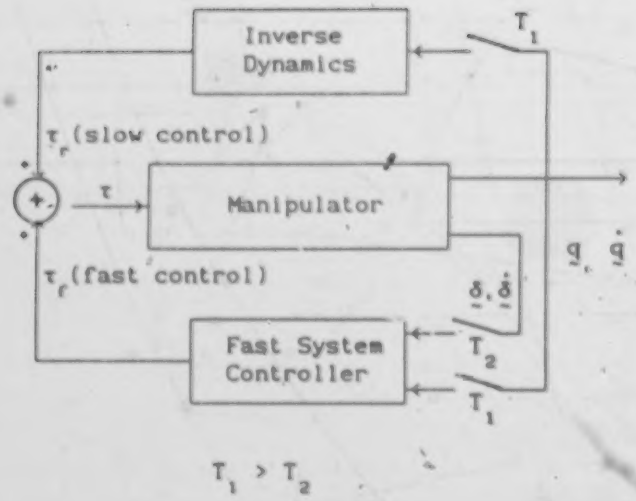


Figure 4.9. Block Diagram of the Multirate Controller..

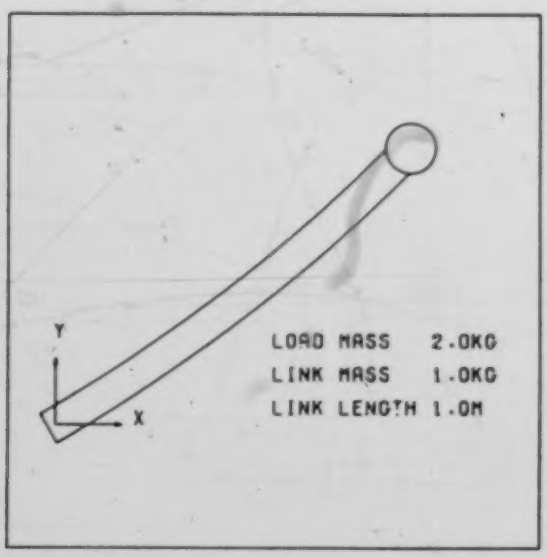


Figure 4.10. Single-link manipulator used in the multirate Example.

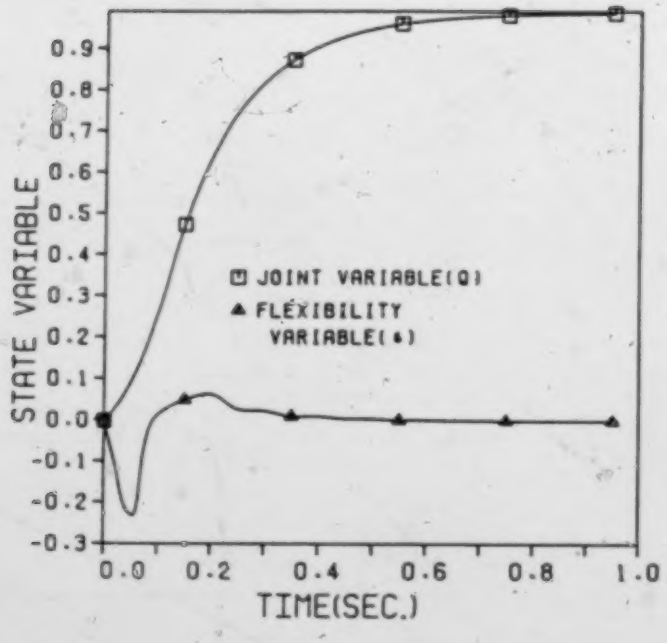


Figure 4.11. Responses of joint angle and flexibility variable under joint controller only. Joint controller operating at 50hz.

introduced into the flexible subsystem model did not prevent instability at this sampling rate. If this damping was not present, instability would have occured at an even higher sampling rate.

The ability to sample the joint angles at a slow rate when a fast controller is added to the system is next shown in Figure 4.12. This graph shows the responses of the joint and flexibility variables when the slow controller is operated at a rate of 20 Hertz and the fast controller is operated at a rate of 100 Hertz. The responses are now almost identical to the responses shown in Figure 4.11. Stability is restored and the flexibility variable is damped out quite rapidly. The decrease in sampling rate of the slow controller that is made possible by the multirate control scheme can be a significant factor in reducing speed requirements of the manipulator control system hardware.

4.10. Conclusion.

The multirate control scheme has been tested on a single link moving in a plane. The success of the scheme in this application is no guarantee that it will work as well when applied to a general, multi-linked manipulator. This merits further investigation but this has not been pursued as part of this thesis. It is possible that in general, rates as high as 50 Hertz might be necessary for control of the rigid subsystem. A multirate, pseudolink composite controller would require inclusion of flexibility variables in the computed torque controller and calculation of the inverse dynamics, including flexibility effects, might be necessary. This is where the Newton-Euler like

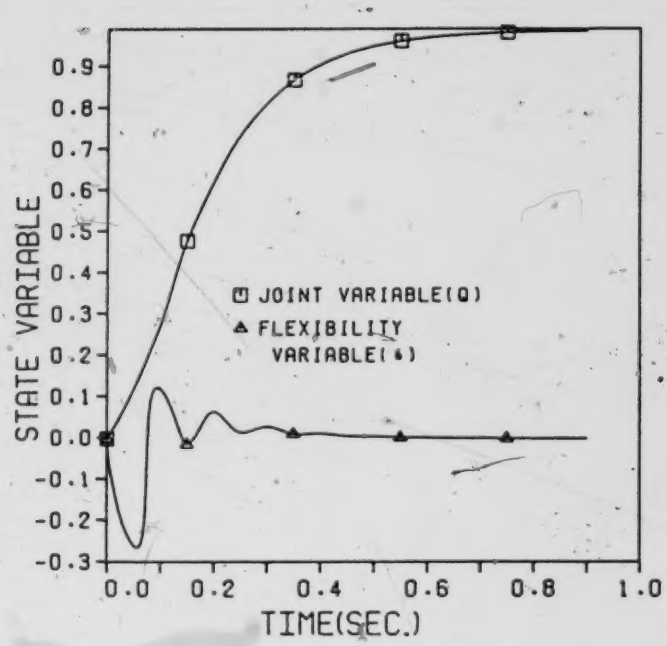


Figure 4.12 Responses of the joint angle and flexibility variable under multirate control: Joint controller operating at 20hz, Flexibility controller operating at 100hz.

algorithms would prove useful. In order to accomodate a sampling rate of 50 Hertz, parallel processing would be necessary. Parallel processing of the assumed modes, inverse dynamics algorithm is described in the next chapter.

CHAPTER 5. RECURRENCE RELATIONS AND PARALLELISM IN FLEXIBLE MANIPULATOR DYNAMICS.

5.1. Introduction.

In order to implement the composite, pseudolink control scheme described in the previous chapter, the control torques must be computable in real time by reasonably priced hardware. The computational requirements of this control system, and ways to meet these requirements, form the subject of this chapter.

5.2 Computational Requirements of the Composite, Pseudolink Controller.

The control torques, τ_c, that are required for composite, pseudolink control of a multilink, flexible manipulator are given by the following equation:

$$\underset{\sim}{\tau}_c = H_{11}\,\underset{\sim}{\nu}(t) + H_{12}\,\ddot{\underset{\sim}{\delta}}(t) + \underset{\sim}{h}_{1,c} \qquad (5\text{-}1)$$

where:

$$\underset{\sim}{\nu}(t) = \underset{\sim}{\nu}_1(t) + \underset{\sim}{\nu}_2(t) \qquad (5\text{-}2)$$

$\underset{\sim}{\nu}(t)$ is the vector of joint accelerations that are needed in order to drive the manipulator pseudolink angles along their desired trajectories without instability of the flexible subsystem of the manipulator. The vector $\underset{\sim}{\nu}_1(t)$ is designed to achieve decoupling of

the joint dynamics from the flexible dynamics, and includes terms that are selected in order to endow the closed-loop, joint dynamical subsystem with convenient, linear properties. The vector $\underset{\sim}{\nu}_2(t)$ is designed to stabilize the flexible subsystem. Its elements decay rapidly to zero. Once these two vectors are determined, substitution of their sum into equation (5-1) yields the required control torques. This equation represents the inverse dynamics algorithm for flexible manipulators that has been presented in Chapter 2 of this thesis. Calculation of the control torques can therefore be performed by that algorithm, with the vector $\underset{\sim}{\nu}(t)$ substituted for the joint accelerations in the algorithm.

The problem, however, is complicated by the fact that the procedure for calculating $\underset{\sim}{\nu}_2(t)$ requires the other submatrices of the inertia matrix H, which is partitioned as follows:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \qquad (5\text{-}3)$$

(The subscript "c" is omitted for brevity.) All the terms of H are therefore needed at some point. It seems that calculation of the control torques is more efficiently done by separately calculating the bias torques $\underset{\sim}{h}$ (= $[\underset{\sim}{h}_1 \mid \underset{\sim}{h}_2]^T$ as defined by equations (4-5a) and (4-5b)), and the inertia matrix H. The bias vector $\underset{\sim}{h}_1$ and the matrix H_{11} are then literally substituted into equation (5-1) to calculate the torques.

The computational requirements for computed torque control of flexible manipulators can therefore be summarized as follows:

1) Calculation of the bias vector, and,

2) Calculation of the inertia matrix H.

Both the bias vector and the inertia matrix need to be re-calculated at regular intervals, perhaps during every sampling interval. Calculation of the bias vectors is performed by the inverse dynamics algorithm with all mention of generalized accelerations omitted, and the inertia matrix is calculated by the forward dynamics algorithm.

We have already seen, the recursive algorithms presented in Chapter 2 involve fewer arithmetic computations than the traditional Lagrangian algorithms. Nontheless, greater processing power than is available from current microprocessor chips is needed to implement the algorithms in real time. One way of obtaining the required computational power is through parallel processing. This approach has been taken by several researchers for achieving the speed needed for real time control of rigid manipulators [55-62, 69]. Recursive equations are suitable for computation using a class of powerful, highly parallel computers, referred to a Single Instruction Multiple Data stream (SIMD) computers [100, 101]. However, the precedence relationships that exist between various terms of recursive equations must be organized in an efficient way so as not to degrade the performance of the computer. Pipelining is a concept that is in widespread use in SIMD computers, and methods for arranging the recursion to make full use of this architectural feature are required.

The pipelining methods of Lee and Chang [62,69] are particularly interesting. They reformulate the Newton-Euler equations into "linear recurrences". In this form, the recursive

equations can be efficiently computed by small, special purpose, SIMD computers. In the sequel, it is shown that the recursive equations in the inverse and forward dynamics algorithms for flexible manipulators can also be reformulated into linear recurrence form for efficient computation by SIMD computers.

5.3 Parallelism in Linear Recurrence Problems.

A linear recurrence problem of size (n+1) can be described as the computation of a quantity X(i) that is given by the following equation:

$$X(i) = a(i) * X(i-1) + b(i), \quad 0 \le i \le n \qquad (5\text{-}4)$$

where "*" and "+" are two associative binary operators, X(0) is not the identity with respect to "*", and a(i) and b(i) are given for all i. If a(i) is the identity with respect to "*", or if b(i) is the identity element with respect to "+", then the problem is a linear, *homogenous* recurrence relation. Otherwise, it is a linear *inhomogenous* recurrence relation.

A simple example of a linear, homogenous recurrence relation is the problem of computing the sum of (n+1) integers $a_0, a_2, \ldots a_n$, where n is a power of 2. This problem can be written in the following form:

$$y_0 = a_0, \qquad (5\text{-}5a)$$

$$y_i = y_{i-1} + a_i, \quad 1 \le i \le n \qquad (5\text{-}5b)$$

A linear, homogenous recurrence problem can be parallelized by

grouping consecutive pairs of terms together, and performing the operation on each pair simultaneously. The operation is then repeated in a similar manner until the final result is obtained as the result of the last stage of the process. This process is called the "recursive doubling" technique. The recursive doubling technique applied to the addition example is depicted by the tree-like structure in Figure 5.1. The leaf nodes represent the integers to be added, and the value at each intermediate node is a "partial sum". The value at the root node is the required final sum. This procedure clearly requires $\log_2(n)$ stages and therefore results in reducing the computational complexity of the summation algorithm from O(n), as it would be when computed on a uniprocessor, to O(log(n)) when computed in parallel using the recursive doubling technique.

A linear inhomogenous recurrence problem can also be evaluated in parallel using the recursive doubling method. The process, however, is more complicated. The reader is referred to Lee and Chang [69] for the algorithm.

The recursive doubling algorithm is susceptible to a "systolic pipelined" design that can be implemented using very large scale integration (VLSI) technology. Systolic pipelining is a technique that has recently been developed for reducing the design cost of special purpose integrated devices [102-104]. Devices with this architecture are characterized by a large number of simple computing elements through which data flows like blood through veins. The computing elements are members of a small set of basic types. The layout of each of these basic building blocks is first optimized, then they are simply replicated a large number of times

Figure 5.1 omitted because of copyright protection.

Figure 5.1. Diagram showing the recursive doubling computational algorithm. (Taken from Reference 62)

as needed. Each computing element communicates with its nearest neighbour only. I/O operations are performed only by boundary elements. Communication problems are therefore avoided. Systolic architectures conform well to the constraints imposed by VLSI technology.

Not all algorithms are susceptible to a systolic design. In general, problems that are "compute bound", that is, problems that involve a greater number of arithmetic operations than I/O operation, can benefit from a systolic pipelined design. Many signal processing algorithms are compute bound and can be efficiently parallelized using the systolic approach.

5.4. Inverse Dynamics of Flexible Manipulators as Linear Recurrences.

As was mentioned previously, it is shown by Lee and Chang [62] that the recursive equations that make up the Newton-Euler, inverse dynamics algorithm for rigid manipulators can be re-written in the form of linear, homogenous recurrence relations. The Newton-Euler, forward dynamics equations can be written partly in the form of linear homogenous and partly in the form of linear inhomogenous recurrence relations. Recursive doubling methods have been proposed for parallelizing both algorithms. In this section, we show that the inverse dynamics algorithm for flexible manipulators can also be written as linear, homgenous recurrence relations. In a later section, we show that the forward dynamics algorithm for flexible manipulators can also be formulated partly as homogenous and partly as inhomogenous linear recurrences. The recursive doubling technique can therefore be employed to design

efficient, systolic architectures for parallelizing the algorithms.

The inverse dynamics algorithm with generalized accelerations omitted is summarized in Table 5.1. The equations that form part of the Newton-Euler, inverse dynamics algorithm for rigid manipulators are indicated. The similarities between the algorithms for rigid and flexible manipulators are striking. There are, however, significant differences. In the first place, the equations that are necessary to calculate flexible link kinematics (equations F1-1 to F1-14) are not present in the algorithm for rigid manipulators. In the second place, equations (F1-16), (F1-18), (F1-20), (F1-20), (F1-21), parts of equations (F1-22) and (F1-23), equations (F1-24) to (F1-26), and equations (F1-29) to (F1-31) are all new to the algorithm for flexible dynamics. Fortunately, these two groups of equations do not destroy the basic recursive nature of the algorithm. The recursiveness of the algorithm is actually exhibited by equations (F1-15) to (F1-20) and equations (F1-27) and (F1-28).

5.4.1 Model for Calculating the Non-Recursive Equations.

Equations (F1-1) to (F1-14) do not form part of the recursive procedure for calculating the torques. However, their results are needed by the recursive equations. Fortunately, these equations do not represent a major computational bottleneck to the overall computational scheme. The most straightforward approach to their incorporation into the global computational picture is to have them calculated before the main recursive procedure commences. Separate arithmetic units can be employed for this purpose. These

Table 5.1.

Summary of Inverse Dynamics Algorithm for Flexible Manipulators, with Generalized accelerations omitted.

Assume: q_i, $\dot{q}_i$, $[\delta_{ik}]$, $[\dot{\delta}_{ik}]$, ${}^{i}p_i^{r\bullet}$, ${}^{i}\hat{s}_i^{r}$, for each i and each k, where indicated.

INITIAL CONDITIONS.

${}^{0}\underset{\sim}{\omega}_0 = {}^{0}\dot{\underset{\sim}{\omega}}_0 = \underset{\sim}{0}$; ${}^{0}\dot{\underset{\sim}{v}}_0 = g$ (gravity vector); N = # links, NM = # modes per link.

INITIAL CALCULATIONS.

For i = 1 to N step 1, DO

{

$$ {}^{i}\dot{\underset{\sim}{\phi}}_i = \sum_{k=1}^{NM} [\dot{\delta}_{ik}]\, \bar{\underset{\sim}{\phi}}_{ik} \qquad \text{(F1-1)} $$

$$ {}^{i}\underset{\sim}{A}_i = \sum_{k=1}^{NM} [\delta_{ik}]\, \bar{\underset{\sim}{A}}_{ik} \qquad \text{(F1-2)} $$

$$ {}^{i}\dot{\underset{\sim}{A}}_i = \sum_{k=1}^{NM} [\delta_{ik}]\, \bar{\underset{\sim}{A}}_{ik} \qquad \text{(F1-3)} $$

$$ {}^{i}\hat{\underset{\sim}{A}}_i = \sum_{k=1}^{NM} [\delta_{ik}]\, \bar{\underset{\sim}{e}}_{ik} \qquad \text{(F1-4)} $$

$$ {}^{i}\dot{\hat{\underset{\sim}{A}}}_i = \sum_{k=1}^{NM} [\delta_{ik}]\, \bar{\underset{\sim}{e}}_{ik} \qquad \text{(F1-5)} $$

Table 5.1 (Cont'd)

$$ {}^{1}\underset{\sim}{\overset{\circ}{h}}_{1} = \begin{bmatrix} 0 \\ \sum_{k=1}^{NH} \delta_{1kz} \bar{h}_{1kz} \\ \sum_{k=1}^{NH} \delta_{1ky} \bar{h}_{1ky} \end{bmatrix} \qquad \text{(F1-6)} $$

$$ {}^{1}\underset{\sim}{\overset{s}{f}}_{1} = \begin{bmatrix} \sum_{k=1}^{NH} \sum_{l=1}^{NH} \delta_{1ky} \delta_{11z} \bar{f}_{1klyz} \\ 0 \\ 0 \end{bmatrix} \qquad \text{(F1-7)} $$

$$ \overset{\circ}{b}_{1} = \sum_{k=1}^{NH} \sum_{l=1}^{NH} \left[\delta_{1ky} \delta_{11y} \bar{b}_{1kly} + \delta^{2}_{1kx} \bar{\phi}_{1kx} \right] \qquad \text{(F1-8)} $$

$$ f_{1ky} = \sum_{l=1}^{NH} \delta_{11y} \bar{f}_{1klyz} \qquad \text{(F1-9)} $$

$$ f_{1kz} = \sum_{l=1}^{NH} \delta_{11z} \bar{f}_{1klzy} \qquad \text{(F1-10)} $$

$$ {}^{1}\underset{\sim}{G}_{1ky} = {}^{1}\underset{\sim}{G}^{rf}_{1ky} + \sum_{k=1}^{NH} \delta_{11y} \bar{G}_{1kly} \qquad \text{(F1-11)} $$

$$ K_{1kx} = \sum_{l=1}^{NH} \delta_{11x} \bar{K}_{11x} \qquad \text{(F1-12)} $$

Table 5.1 (Cont'd)

$$K_{iky} = \sum_{l=1}^{NN} \delta_{ily} \bar{K}_{ily} \qquad \text{(F1-13)}$$

$$K_{ikz} = \sum_{l=1}^{NN} \delta_{ilz} \bar{K}_{ilz} \qquad \text{(F1-14)}$$

NEXT 1;

/* Note: All symbols with overscores are constant */

} /* END INITIAL CALCULATIONS */

FORWARD RECURSION.

For 1 = 1 to N, step 1, DO:

{

Calculate ${}^{1}A_{1-1}$ and E_1^{rf}.

$$ {}^{1}\underset{\sim}{\omega}_1^r = {}^{1}A_{1-1}\left[{}^{1-1}\underset{\sim}{\omega}_{1-1} + \underset{\sim}{z}_0 \dot{q}_1\right] \qquad \text{(F1-15)}$$

(Same as for Rigid Manipulator)

$$ {}^{1}\underset{\sim}{\omega}_1 = E_1^{rf}\left[{}^{1}\underset{\sim}{\omega}_1^r + {}^{1}\underset{\sim}{\phi}_1\right] \qquad \text{(F1-16)}$$

$$ {}^{1}\dot{\underset{\sim}{\omega}}_1^r = {}^{1}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{\omega}}_{1-1} + {}^{1-1}\underset{\sim}{\omega}_{1-1} \times \underset{\sim}{z}_0\dot{q}_1\right] \qquad \text{(F1-17)}$$

(Same as for Rigid Manipulator)

$$ {}^{1}\dot{\underset{\sim}{\omega}}_1 = E_1^{rf}\left[{}^{1}\dot{\underset{\sim}{\omega}}_1^r + {}^{1}\underset{\sim}{\omega}_1^r \times {}^{1}\underset{\sim}{\phi}_1\right] \qquad \text{(F1-18)}$$

Table 5.1 (Cont'd)

$$ {}^{1}\dot{\underset{\sim}{V}}_{1}^{r} = {}^{1}A_{1-1}\left[{}^{1-1}\dot{\underset{\sim}{V}}_{1-1}\right] + {}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} \times {}^{1}\underset{\sim}{p}_{1}^{r*} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times ({}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\underset{\sim}{p}_{1}^{r*}) $$

(F1-19)

(Same as for Rigid Manipulator)

$$ {}^{1}\dot{\underset{\sim}{V}}_{1} = E_{1}^{r}\left[{}^{1}\dot{\underset{\sim}{V}}_{1}^{r} + {}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} \times {}^{1}\underset{\sim}{\Delta}_{1} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times ({}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\underset{\sim}{\Delta}_{1}) + 2({}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\dot{\underset{\sim}{\Delta}}_{1})\right] \quad \text{(F1-20)} $$

$$ {}^{1}\ddot{\underset{\sim}{V}}_{1}^{r} = {}^{1}\dot{\underset{\sim}{V}}_{1}^{r} + {}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} \times {}^{1}\underset{\sim}{s}_{1}^{r} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times ({}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\underset{\sim}{s}_{1}^{r}) \quad \text{(F1-21)} $$

(Same as for Rigid Manipulator)

$$ {}^{1}\underset{\sim}{F}_{1} = \left[M_{1}\,{}^{1}\ddot{\underset{\sim}{V}}_{1}^{r} + {}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} \times {}^{1}\dot{\underset{\sim}{\Delta}}_{1} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times \left[{}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\dot{\underset{\sim}{\Delta}}_{1}\right] + 2\left[{}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}\dot{\underset{\sim}{\Delta}}_{1}\right]\right] \quad \text{(F1-22)} $$

(Except for the terms in Δ, same as for Rigid Manipulator)

$$ {}^{1}\underset{\sim}{N}_{1} = \left[{}^{1}I_{1}^{f}\,{}^{1}\dot{\underset{\sim}{\omega}}_{1}^{r} + {}^{1}\dot{I}_{1}^{f}\,{}^{1}\underset{\sim}{\omega}_{1}^{r} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times {}^{1}I_{1}^{f}\,{}^{1}\underset{\sim}{\omega}_{1}^{r} + {}^{1}\dot{\underset{\sim}{\Delta}}_{1} \times {}^{1}\ddot{\underset{\sim}{V}}_{1}^{r} + {}^{1}\underset{\sim}{\omega}_{1}^{r} \times \left[{}^{1}\overset{*}{\underset{\sim}{h}}_{1} + {}^{1}\overset{s}{\underset{\sim}{f}}_{1}\right] \right] \quad \text{(F1-23)} $$

(Except for terms that include ${}^{1}\overset{*}{\underset{\sim}{h}}_{1}$, ${}^{1}\overset{s}{\underset{\sim}{f}}_{1}$, ${}^{1}\dot{I}_{1}^{f}$ and ${}^{1}\dot{\underset{\sim}{\Delta}}_{1}$, same as for rigid Manipulator)

Table 5.1 (Cont'd)

For i = N down to 1, step -1, DO:

{

$$ {}^{i}\underset{\sim}{f}_{i} = E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{f}_{i+1} + {}^{i}\underset{\sim}{F}_{i} \qquad \text{(F1-27)} $$

(Except for E_i^{fr}, same as for rigid Manipulator)

$$ {}^{i}\underset{\sim}{n}_{i} = E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{n}_{i+1} + \left[{}^{i}\underset{\sim}{p}_{i}^{r^{*}} + {}^{i}\underset{\sim}{\Delta}_{i}\right] \times E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{f}_{i+1} + \left[{}^{i}\underset{\sim}{p}_{i}^{r^{*}} + {}^{i}\underset{\sim}{\hat{s}}_{i}^{r}\right] \times {}^{i}\underset{\sim}{F}_{i} + {}^{i}\underset{\sim}{N}_{i} \qquad \text{(F1-28)} $$

(Except for E_i^{fr} and ${}^{i}\underset{\sim}{\Delta}_{i}$, same as for rigid Manipulator)

/* Calculate Torques due to Flexibility */

For k = 1 to NM, step 1, DO:

{

$$ \tau_{ikx} = t_{ikx} + {}^{i}\underset{\sim}{\Phi}_{ikx} \cdot E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{n}_{i+1} \qquad \text{(F1-29)} $$

$$ \tau_{iky} = t_{iky} + {}^{i}\underset{\sim}{\Phi}_{iky} \cdot E_{i}^{fr}\ {}^{i}A_{iky}\ {}^{i+1}\underset{\sim}{n}_{i+1} + {}^{i}\underset{\sim}{\Delta}_{iky} \cdot E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{f}_{i+1} \qquad \text{(F1-30)} $$

$$ \tau_{ikz} = t_{ikz} + {}^{i}\underset{\sim}{\Phi}_{ikz} \cdot E_{i}^{fr}\ {}^{i}A_{ikz}\ {}^{i+1}\underset{\sim}{n}_{i+1} + {}^{i}\underset{\sim}{\Delta}_{ikz} \cdot E_{i}^{fr}\ {}^{i}A_{i+1}\ {}^{i+1}\underset{\sim}{f}_{i+1} \qquad \text{(F1-31)} $$

NEXT k;

} /* END INNER LOOP */

Table 5.1 (Cont'd)

```
    /* CALCULATE JOINT TORQUE */

    τ_j = z_0 · ^{i-1}A_i ^i n_i                                    (F1-32)

} /***  END MAKING ALL TORQUES  ***/

**************************END TABLE 5.1**************************
```

units can themselves possess pipelined architectures for performing fast floating point operations but this pipelining exists at a lower level than what is considered in implementing the recursive equations.

Conceptually, one can have a separate processor for each of equations (F1-1) to (F1-14). Each processor performs an inner product operation for each of the x, y and z directions simultaneously. In otherwords, for the y-direction, say, the processor computes the following operation:

$$\Delta_{1y} = \begin{bmatrix} \delta_{11y} \\ \delta_{12y} \\ \vdots \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} \bar{\Delta}_{11y} \\ \bar{\Delta}_{12y} \\ \vdots \\ \vdots \end{bmatrix} \qquad (5\text{-}6)$$

This operation can be performed by the processing elements shown in Figure 5.2 in just a few clock cycles. Similar operations are performed simultaneaouly for the x and z components of the 3-dimensional vector ${}^{1}\underset{\sim}{\Delta}_{1}$ represented by equation (F1-2). The same arithmetic unit is used to compute the vectors for all flexible links. In this way, hardware requirements are kept low.

All the vectors indicated as the results of equations (F1-1) to (F1-14) are calculated simultaneously. This is possible, since all data required for the calculations are available at the start of the computational procedure. VLSI implementation of the processors is also possible due to the similarity of the processing elements. These can all be constructed from two basic building blocks, namely, an inner product unit and a summer. VLSI

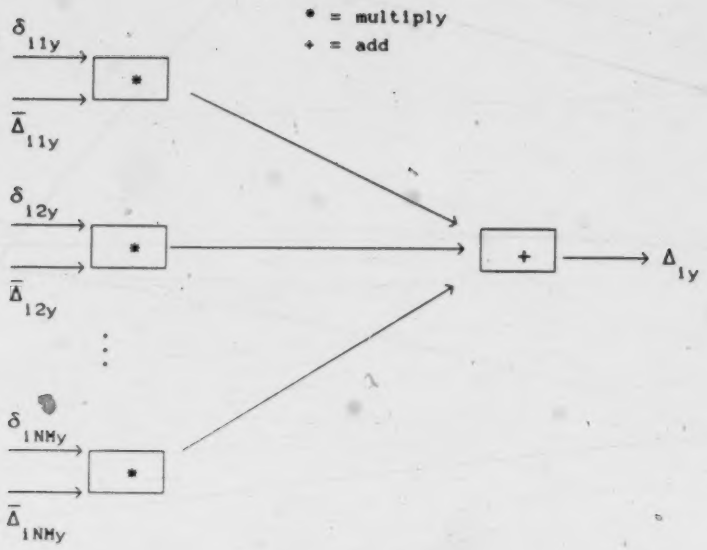


Figure 5.2. Conceptual Model and Processing Elements for Performing the Non-Recursive Computations.

design can therefore concentrate on optimizing the layout of these blocks, and the layout replicated over several chips.

5.4.2. Parallel Algorithm for Calculating the Recursive Equations.

The major computational bottleneck of the inverse dynamics algorithm for flexible manipulators is presented by equations (F1-15) to (F1-32). Precedence relationships exist between the various terms of these equations, hence efficient methods for computing them in parallel, using SIMD computers, need to be devised. In the form shown in Table 5-1, the precedence relationships are not represented by linear recurrences. However,

by first referring all vector quantities to base coordinates, and then combining the following pairs of equations: (F1-15, F1-16), (F1-17, F1-18), and (F1-19, F1-20), all the recursive equations can be written in linear, homogenous recurrence form.

The resulting procedure for evaluating the recursive equations as a linear, homogenous recurrence problem is given below.

1) Compute the 3x3 rotation matrices, ${}^{0}T_{i}^{r}$, i=1,2,...N, by:

$$ {}^{0}T_{i}^{r} = {}^{0}T_{i-1}^{r}\, E_{i}^{fr}\, {}^{i-1}A_{i} \qquad (5\text{-}7) $$

2) Compute the vectors p_i^{r*}, $\hat{s}_i^r$, $\mathring{\phi}_i$, $\underset{\sim}{\Delta}_i$, $\mathring{\underset{\sim}{\Delta}}_i$, $\hat{\underset{\sim}{\Delta}}_i$, $\hat{\mathring{\underset{\sim}{\Delta}}}_i$ $\underset{\sim}{z}_i$, $\mathring{\underset{\sim}{h}}_i$, $\mathring{\underset{\sim}{f}}_i$, i=1,2,...N, by the following equations:

$$ \underset{\sim}{z}_i = {}^{0}T_{i}^{r}\, \underset{\sim}{z}_0, \quad \underset{\sim}{z}_0 = [0 \;\; 0 \;\; 1]^T \qquad (5\text{-}8) $$

$$ p_i^{r*} = {}^{0}T_{i}^{r}\, {}^{i}p_i^{r*} \qquad (5\text{-}9) $$

The reset of the vectors are calculated by equations of the form of (5-9). This step refers all vectors to base coordinates.

3) Compute

$$ \underset{\sim}{b}_i = \underset{\sim}{z}_{i-1}\, \dot{q}_i + \mathring{\underset{\sim}{\phi}}_{i-1} \qquad (5\text{-}10) $$

and

$$\underset{\sim}{\omega}_i^r = \underset{\sim}{\omega}_{i-1}^r + \underset{\sim}{b}_i \qquad (5\text{-}11)$$

4) Compute

$$\underset{\sim}{b}_i = \underset{\sim}{\omega}_{i-1}^r \times \left[\underset{\sim}{z}_{i-1}\dot{q}_i + \overset{\circ}{\underset{\sim}{\phi}}_{i-1} \right] \qquad (5\text{-}12)$$

and

$$\dot{\underset{\sim}{\omega}}_i^r = \dot{\underset{\sim}{\omega}}_{i-1}^r + \underset{\sim}{b}_i \qquad (5\text{-}13)$$

5) Compute

$$\begin{aligned} \underset{\sim}{b}_i = {} & \dot{\underset{\sim}{\omega}}_i^r \times \underset{\sim}{p}_i^{r*} + \underset{\sim}{\omega}_i^r \times (\underset{\sim}{{}^1\omega}_i^r \times \underset{\sim}{{}^1p}_i^{r*}) \\ & + \dot{\underset{\sim}{\omega}}_{i-1}^r \times \underset{\sim}{\Delta}_{i-1} + \underset{\sim}{\omega}_{i-1}^r \times (\underset{\sim}{\omega}_{i-1}^r \times \underset{\sim}{\Delta}_{i-1}) \\ & + 2(\underset{\sim}{{}^1\omega}_i^r \times \overset{\circ}{\underset{\sim}{{}^1\Delta}}_i) \Big] \end{aligned} \qquad (5\text{-}14)$$

and

$$\dot{\underset{\sim}{v}}_i^r = \dot{\underset{\sim}{v}}_{i-1}^r + \underset{\sim}{b}_i \qquad (5\text{-}15)$$

6) Compute

$$\hat{\dot{\underset{\sim}{v}}}_i^r = \dot{\underset{\sim}{v}}_i^r + \dot{\underset{\sim}{\omega}}_i^r \times \hat{\underset{\sim}{s}}_i^r + \underset{\sim}{\omega}_i^r \times (\underset{\sim}{\omega}_i^r \times \hat{\underset{\sim}{s}}_i^r) \qquad (5\text{-}16)$$

7) Compute

$$\underset{\sim}{F}_i = \left[M_i \overset{\circ}{\underset{\sim}{v}}^r_i + \dot{\underset{\sim}{\omega}}^r_i \times \hat{\underset{\sim}{\Delta}}_i + \underset{\sim}{\omega}^r_i \times \left(\underset{\sim}{\omega}^r_i \times \hat{\underset{\sim}{\Delta}}_i \right) + 2\left(\underset{\sim}{\omega}^r_i \times \hat{\underset{\sim}{\delta}}_i \right) \right] \qquad (5\text{-}17)$$

8) Compute

$$\underset{\sim}{N}_i = \left[I^f_i \dot{\underset{\sim}{\omega}}^r_i + \overset{\circ}{I}^f_i \underset{\sim}{\omega}^r_i + \underset{\sim}{\omega}^r_i \times I^f_i \, {}^i\underset{\sim}{\omega}^r_i + \hat{\underset{\sim}{\Delta}}_i \times \overset{\circ}{\underset{\sim}{v}}^r_i + \underset{\sim}{\omega}^r_i \times \left(\overset{\circ}{\underset{\sim}{h}}_i + \overset{\circ}{\underset{\sim}{f}}_i \right) \right] \qquad (5\text{-}18)$$

9) Compute "INNER LOOP" as in Table 5.1.

10) Compute

$$\underset{\sim}{f}_i = \underset{\sim}{f}_{i+1} + \underset{\sim}{F}_i \qquad (5\text{-}19)$$

11) Compute

$$\underset{\sim}{b}_i = \left(\underset{\sim}{p}^{r*}_i + \underset{\sim}{\Delta}_i \right) \times \underset{\sim}{f}_{i+1} + \left(\underset{\sim}{p}^{r*}_i + \hat{\underset{\sim}{s}}^r_i \right) \times \underset{\sim}{F}_i + \underset{\sim}{N}_i \qquad (5\text{-}20)$$

and

$$\underset{\sim}{n}_i = \underset{\sim}{n}_{i+1} + \underset{\sim}{b}_i \qquad (5\text{-}21)$$

12) Compute

$$\tau_{ikx} = t_{ikx} + \underset{\sim}{\Phi}_{ikx} \cdot \underset{\sim}{n}_{i+1} \qquad (5\text{-}22)$$

$$\tau_{iky} = t_{iky} + \underset{\sim}{\Phi}_{iky} \cdot \underset{\sim}{n}_{i+1} + \underset{\sim}{\Delta}_{iky} \cdot \underset{\sim}{f}_{i+1} \qquad (5\text{-}23)$$

$$\tau_{ikz} = t_{ikz} + \underset{\sim}{\Phi}_{ikz} \cdot \underset{\sim}{n}_{i+1} + \underset{\sim}{\Delta}_{ikz} \cdot \underset{\sim}{f}_{i+1} \qquad (5\text{-}24)$$

$$\text{for } k = 1,2,\ldots NM$$

13) Compute

$$\tau_j = \underset{\sim}{z}_{i-1} \cdot \underset{\sim}{n}_i \qquad (5\text{-}25)$$

It is seen that all the recursive equations, namely, equations (5-11), (5-13), (5-15), (5-19) and (5-21), are now in the form of linear, homogenous recurrence relations. The other equations are not recursive and can be evaluated by simple parallel computations. The computation of the recursive equations can now be performed by employing the recursive doubling algorith. Implementation on SIMD computers is can now be efficiently done.

5.5. Forward Dynamics of Flexible Manipulators as Linear Recurrences.

As mentioned previously, Lee and Chang [69] have shown that some of the recursive equations that form part of the Newton-Euler, forward dynamics algorithm can be written in the form of linear, homogenous recurrences. In fact, all except one of them can be wrritten in this form. The remaining recursive equation can be written in the form of a linear, inhomogenous equation. Both types of recurrence problems can be evaluated by using the recursive doubling algorithm. This results in an O(log N) computational time complexity compared with an O(N) time

complexity for the non-parallel method. In this section, the recursive equations that comprise the algorithm for calculating the elements of the inertia matrix, H, of a flexible manipulator, are re-written in linear recurrence form. This requires referring all vectors and inertia tensors to base coordinates. The resulting computational procedure is given in Table 5.2. In these equations, left superscripts are omitted for brevity. All vectors and inertia tensors, however, are understood to be referred to base coordinates.

Table 5.2 shows that equations (F2-1), (F2-3), (F2-15), (F2-16), (F2-19), (F2-24), (F2-25), and F2-31), are all in linear, homogenous recurrence form. Similar equations exist in the modules for computing columns that correspond to madal variables in the y and x directions. These equations are not shown in Table 5.2, but they possess the same recurrence properties as those that are shown.

In the whole algorithm, only equation (F2-2) is in linear inhomogenous form. Equations (F2-18), and (F2-30) are simple assignments. The rest of the equations are not recursive and can therefore be evaluated by simple parallel computations.

5.6. Conclusion.

In this chapter, the inverse and forward dynamics algorithms for flexible manipulators that have been developed in this thesis have been shown to possess convenient, linear recursive properties. These properties allow for efficient parallelization of the algorithms by use of the method of recursive doubling. The computational time complexity of the algorithms becomes of order

Table 5.2

Summary of the Linear Recursive Algorithm for Calculating the Inertia Matrix H.

Assume: q_i, $[\delta_{ik}]$, p_i^{r*}, $\hat{s}_i^r$, m_i, I_i^f, $\underset{\sim}{\Delta}_i$, $\hat{\underset{\sim}{\Delta}}_i$, $\underset{\sim}{\Delta}_{ikv}$, $\underset{\sim}{\Phi}_{ikv}$, for all $i = 1,2, \ldots N$, $k = 1,2,\ldots NM$, where indicated.

N = number of links, NM = number of modes per link.

INITIAL CONDITIONS.

M_{N+1} = payload mass.

$\underset{\sim}{c}_{N+1}$ = position vector of payload from the end-point of the N'th link,

$\underset{\sim}{\mathcal{E}}_{N+1}$ = inertia tensor of payload about the end-point of the N'th link.

RECURSION FROM MANIPULATOR TIP DOWN TO ITS BASE

For i = N down to 1, step -1, DO:

{

$$M_i = M_{i+1} + m_i, \qquad \text{(F2-1)}$$

$$\underset{\sim}{c}_i = \left[\frac{M_{i+1}}{M_i}\right] \underset{\sim}{c}_{i+1} + \frac{1}{M_i}\left\{ m_i\left[p_i^{r*} + \hat{\underset{\sim}{s}}_i^r\right] + \hat{\underset{\sim}{\Delta}}_i + M_{i+1}\left[p_i^{r*} + \underset{\sim}{\Delta}_i\right]\right\} \qquad \text{(F2-2)}$$

$$\underset{\sim}{\mathcal{E}}_i = \underset{\sim}{\mathcal{E}}_{i+1} + M_{i+1}\,F(\underset{\sim}{X}_1) + I_i^f - F(\hat{\underset{\sim}{\Delta}}_i)/m_i + m_i\,F(\underset{\sim}{X}_2) \qquad \text{(F2-3)}$$

where $F(\underset{\sim}{X}) = (\underset{\sim}{X} \cdot \underset{\sim}{X})I - \underset{\sim}{X}\,\underset{\sim}{X}^T$ (F2-4)

Table 5.2 (Cont'd)

$$\underset{\sim}{X}_1 = \underset{\sim}{c}_{i+1} + \underset{\sim}{p}_i^{r*} + \underset{\sim}{c}_i \qquad \text{(F2-5)}$$

$$\underset{\sim}{X}_2 = \underset{\sim}{p}_i + \hat{s}_i + \frac{\hat{\Delta}_i}{m_i} - \underset{\sim}{c}_i \qquad \text{(F2-6)}$$

CALCULATE COLUMNS OF H CORRESPONDING TO FLEXIBILITY VARIABLES:

for k = NM down to 1, step -1, DO:

{

CALCULATE COLUMN OF H CORRESPONDING TO FLEXIBILITY VARIABLE IN THE Z-DIRECTION:

$$f_{i+1:\delta ikz} = M_{i+1}\left[\underset{\sim}{\Delta}_{ikz} + \underset{\sim}{\Phi}_{ikz} \times \underset{\sim}{c}_{i+1}\right] \qquad \text{(F2-7)}$$

$$n_{i+1:\delta ikz} = \mathcal{E}_{i+1}\,\underset{\sim}{\Phi}_{ikz} + \underset{\sim}{c}_{i+1} \times f_{i+1:\delta ikz} \qquad \text{(F2-8)}$$

CALCULATE DIAGONAL TERM:

PZ = N + 3*(i-1)*NM + 3*k; QZ = PZ;

$$H[PZ][PZ] = b_{ikkz} + \underset{\sim}{\Phi}_{ikz} \cdot n_{i+1:\delta ikz} + \underset{\sim}{\Delta}_{ikz} \cdot f_{i+1:\delta ikz} \qquad \text{(F2-9)}$$

/* y-term */

QZ = QZ-1;

$$H[PZ][QZ] = b_{ikky} + \underset{\sim}{\Phi}_{iky} \cdot n_{i+1:\delta ikz} + \underset{\sim}{\Delta}_{iky} \cdot f_{i+1:\delta ikz} \qquad \text{(F2-10)}$$

/* x-term */

QZ = QZ-1;

$$H[PZ][QZ] = b_{ikkx} + \underset{\sim}{\Phi}_{ikx} \cdot n_{i+1:\delta ikz} \qquad \text{(F2-11)}$$

Table 5.2 (Cont'd)

CALCULATE ELEMENTS IN THIS COLUMNS
CORRESPONDING TO OTHER MODES IN THE i'^{th} LINK:

for l = k-1 down to 1, step -1, DO:

{

/* z-term */

QZ = QZ-1;

H[PZ][QZ] = H[QZ][PZ]

$$= b_{ikiz} + \underset{\sim}{\Phi}_{iiz} \cdot n_{i+1:\delta ikz} + \underset{\sim}{\Delta}_{iiz} \cdot f_{i+1:\delta ikz} \qquad \text{(F2-12)}$$

/* y-term */

QZ = QZ-1;

H[PZ][QZ] = H[QZ][PZ]

$$= b_{ikiy} + \underset{\sim}{\Phi}_{iiy} \cdot n_{i+1:\delta ikz} + \underset{\sim}{\Delta}_{iiy} \cdot f_{i+1:\delta ikz} \qquad \text{(F2-13)}$$

/* x-term */

QZ = QZ-1;

H[PZ][QZ] = H[QZ][PZ]

$$= b_{ikix} + \underset{\sim}{\Phi}_{iix} \cdot n_{i+1:\delta ikz} \qquad \text{(F2-14)}$$

NEXT l;

} /* End loop l */

Table 5.2 (Cont'd)

CALCULATE ELEMENT IN THIS COLUMN CORRESPONDING TO THE $1'^{th}$ JOINT VARIABLE:

$$\underset{\sim}{f}_{1:\delta 1kz} = \underset{\sim}{f}_{1+1:\delta 1kz} + \underset{\sim}{e}_{1kz} \qquad \text{(F2-15)}$$

$$\underset{\sim}{n}_{1:\delta 1kz} = \underset{\sim}{n}_{1+1:\delta 1kz} + (p_1^{r^*} + \underset{\sim}{\Delta}_1) \times \underset{\sim}{f}_{1+1:\delta 1kz} + (p_1^{r^*} + \hat{\underset{\sim}{s}}_1^r) \times \underset{\sim}{e}_{1kz} + \underset{\sim}{h}_{1kz} + \underset{\sim}{f}_{1kkz} \qquad \text{(F2-16)}$$

$$H[1][PZ] = H[PZ][1] = \underset{\sim}{z}_{1\phi_1} \cdot \underset{\sim}{n}_{1:\delta 1kz} \qquad \text{(F2-17)}$$

CALCULATE ELEMENTS IN THIS COLUMN CORRESPONDING TO MODES AND JOINT VARIABLES FOR $j < 1$:

For j = 1-1 down to 1, step -1, DO:

{

$$\underset{\sim}{f}_{j:\delta 1kz} = \underset{\sim}{f}_{j+1:\delta 1kz} \qquad \text{(F2-18)}$$

$$\underset{\sim}{n}_{j:\delta 1kz} = \underset{\sim}{n}_{j+1:\delta 1kz} + (p_j^{r^*} + \underset{\sim}{\Delta}_j) \times \underset{\sim}{f}_{j+1:\delta 1kz} \qquad \text{(F2-19)}$$

/* Modes first */

for 1 = MN down to 1, step -1, DO:

{

QZ = QZ - 1;

/* z-term */

H[PZ][QZ] = H[QZ][PZ]

$$= \underset{\sim}{\Phi}_{j1z} \cdot \underset{\sim}{n}_{j+1:\delta 1kz} + \underset{\sim}{\Delta}_{j1z} \cdot \underset{\sim}{f}_{j+1:\delta 1kz} \qquad \text{(F2-20)}$$

QZ = QZ - 1;

Table 5.2 (Cont'd)

/* y-term */

H[PZ][QZ] = H[QZ][PZ]

$$= \underset{\sim}{\Phi}_{j1y} \cdot \underset{}{n}_{j+1:\delta ikz} + \underset{\sim}{\Delta}_{j1y} \cdot \underset{}{f}_{j+1:\delta ikz} \qquad \text{(F2-21)}$$

QZ = QZ - 1;

/* x-term */

H[PZ][QZ] = H[QZ][PZ]

$$= \underset{\sim}{\Phi}_{j1x} \cdot \underset{}{n}_{j+1:\delta ikz} \qquad \text{(F2-22)}$$

NEXT l;

} /* End modes */

/* Joint term */

H[j][PZ] = H[PZ][j]

$$= \underset{\sim}{z}_{i-1} \cdot \underset{\sim}{n}_{j:\delta ikz} \qquad \text{(F2-23)}$$

NEXT j;

} /* End all links j < i */

END MAKING THIS COLUMN.

CALCULATE COLUMN OF H CORRESPONDING TO FLEXIBILITY VARIABLE IN THE Y-DIRECTION:

{

Repeat a proceedure similar to that represented by equations (F2-7) to (F2-23).

}

Table 5.2 (Cont'd)

CALCULATE COLUMN OF H CORRESPONDING TO FLEXIBILITY VARIABLE IN THE X-DIRECTION:

{

Repeat a proceedure similar to that represented by equations (F2-7) to (F2-23).

}

NEXT k;

} /* End modes k */

END MAKING COLUMNS CORRESPONDING TO FLEXIBILITY VARIABLES.

NOW, MAKE COLUMN CORRESPONDING TO THE i'th JOINT VARIABLE.

$$\underset{\sim}{f}_i = \underset{\sim}{f}_{i+1} + \underset{\sim}{z}_{i-1} \times M_i \underset{\sim}{c}_i \qquad \text{(F2-24)}$$

$$\underset{\sim}{n}_i = \underset{\sim}{n}_{i+1} + \varepsilon_i \underset{\sim}{z}_{i-1} + \underset{\sim}{c}_i \times (\underset{\sim}{z}_{i-1} \times M_i \underset{\sim}{c}_i) \qquad \text{(F2-25)}$$

/* Diagonal element */

$$H[i][i] = z_{i-1} \cdot n_i \qquad \text{(F2-26)}$$

/* Elements of this column corresponding to modal variables for joint j < i */

for j = i-1 down to 1, step -1, DO:

P = N + 3*j*NM;

/* Modal variables */

for l = NM down to 1, step -1, DO:

{

/* z-term */

H[i][P] = H[P][i]

$$= \underset{\sim}{\Phi}_{jlz} \cdot n_{j+1} + \underset{\sim}{\Delta}_{jlz} \cdot f_{j+1} \qquad \text{(F2-27)}$$

Table 5.2 (Cont'd)

```
            P = P-1;
            /* y-term */
            H[i][P] = H[P][i]
                    = Φ_jly · n_j+1 + Δ_jly · f_j+1          (F2-28)

            P = P-1;
            /* x-term */
            H[i][P] = H[P][i]  = Φ_jly · n_j+1                (F2-29)

            NEXT l;
        } /* End modal variables */
        /* Joint term */
        f_j  =  f_j+1                                         (F2-30)
        n_j  =  n_j+1  +  (p_j^r* + Δ_j) x f_j+1              (F2-31)

        H[i][j] = H[j][i] = z_j-1 · n_i                       (F2-32)

        NEXT j;
    } /* end loop j */
    NEXT i;
} /* End loop i */
END ALL LINKS.
********************************* END ALGORITHM ***********************
```

(log N). This represents a significant speed-up of the algorithms over their non-parallel evaluation. In fact, the lower bound on computational time is achieved.

The parallel algorithms are suitable for implementation using VLSI technology. Systolic pipelined approaches seem to present a framework around which special purpose computational structures can be designed for their actual implementation. This topic is not pursued any further in this thesis, but it appears to be an area for further research that could prove quite fruitful.

CHAPTER 6. SUMMARY AND PROJECTIONS FOR FURTHER RESEARCH.

6.1. Highlights of Issues Addressed in the Thesis.

Several issues, related to modeling and control of flexible robot manipulators, have been addressed in this thesis. The research project was motivated by the need for lightweight arms that would require less bulky actuator systems and use less energy. The advantages of lightweight arms are presented in Chapter 1. These arms would, of necessity, be made from links that tend to flex as they move. This link flexibility makes the design of control systems that drive the joint actuators a very difficult process.

The problems encountered in designing accurate control systems for flexible manipulators arise mainly because of the complicated dynamics of a multilink, flexible arm that moves in three dimensions (a spatial arm). Modeling distributed link flexibility, and the coupling effects that the links exert on each other, may involve solving complex partial differential equations. Fortunately, this is not necessary in most cases. Discretization of the dynamic equations is possible by use of two methods: (1) the assumed modes method, and (2) the finite element method. These two approaches are used in flexible manipulator modeling in Chapters 2 and 3.

The computational algorithms presented in Chapters 2 and 3 represent the first time that Newton-Euler-like algorithms that use modal or finite element methods have been derived for flexible

manipulators. The Newton-Euler algorithm for calculating rigid manipulator inverse dynamics is well known for its computational efficiency and its linear recursive properties. Huang and Lee [105] extended the Newton-Euler method to flexible manipulators by dividing a link into several sublinks and representing each subjoint by a mass-spring model. This method requires a large number of sublinks for reasonable accuracy. The inverse dynamics algorithm derived in Chapter 2, using the modal approach, involves far fewer degrees of freedom for similar accuracy. Furthermore, the algorithm is very similar to the Newton-Euler algorithm and is also computationally very efficient. An efficient forward dynamics algorithm was also presented in Chapter 2, using the modal approach.

Algorithms similar to those derived in Chapter 2 are presented in Chapter 3. In this chapter, the finite element approach is used to model flexibility. The finite element algorithms involve a much larger number of degrees of freedom than the assumed modes algorithms for similar accuracy. However, they incorporate the boundary conditions at the joints in a more accurate manner and hence do not suffer from the uncertainty in choosing mode shapes that plagues the modal approach. It is shown (in Chapter 3 also), however, that when clamped-free mode shapes are used to model flexibility in the algorithms, simulation results of a manipulator are similar to the results obtained when the finite element algorithms are used for the simulation. Also, mode shapes estimated from the finite element method are found to be similar to the clamped-free mode shapes. It seems, then, that the modal algorithms of Chapter 2 model the dynamics of flexible

manipulators with sufficient accuracy for control and simulation purposes.

In Chapter 4, a new method of flexible manipulator control is presented. Th is a "computed torque" method that is based on decomposing the manipulator dynamic equations into two sets. One set describes that joint motion and the other set describes the vibratory motion due to flexibility. A composite control vector is derived that drives the "psedolink angles" along predetermined trajectories. The major advantage of this approach is that it is systematic and general, and can be applied to spatial manipulators. Simulation results of a three-link, spatial manipulator, confirms this. The pseudolink angles are defined in such a way that desired trajectories can be planned off-line, based on desired tip motion. Pseudolink-angle tracking implies tip-trajectory tracking.

The major disadvantage of the composite, pseudolink control system is its high computational requirements. These requirements are discussed in Chapter 5. Implementing the control system requires real-time calculation of the inverse and forward dynamic equations. It is shown in that chapter that both the inverse and the forward dynamics algorithms for flexible manipulators that have been derived in Chapter 2 can be efficiently parallelized. This is done by re-writing the recursive equations in the form of linear recurrence relations. The recurrence relations are mostly of the homogenous type, with just one of them being of the inhomogenous type. All of them, however, can be evaluated in parallel using the recursive doubling technique. This makes it possible to efficiently compute the recursive equations on SIMD

computers.

6.2. Projections for Further Research.

Although some major issues have been successfully addressed in this thesis, there are many problems that remain to be solved before flexible manipulator control becomes a practical art. The problems lie mainly in implementation of advanced control systems such as the composite, pseudolink control system. Detailed models are frequently required by these controllers. The composite, pseudolink controller employs a complete flexible model, but it may work with a less detailed one. In fact, many of the terms that are included in the computational algorithms may have negligible effects on the overall system model. This conclusion arises from the fact that, in most cases, link deflections are relatively small. These deflections are certainly not negligible, but second-order effects may be. If these terms are located and omitted from the algorithms, simplification of the computational procedure may result. Such a project is certainly worth undertaking, and could form a basis for further research.

Another area of possibly fruitful research is the actual implementation in VLSI of the recursive inverse and forward dynamics algorithms for flexible manipualtors. In Chapter 5, it is shown that efficient parallel implementation of the algorithms is possible, and a systolic design approach is suggested. An actual design, however, was not performed as part of the current research project. Such a design is a substantial undertaking that can comprise a Ph.D thesis. This would involve selection of a small number of basic processing elements that would be replicated many times as needed. Number representations would have to be chosen

(floating point vs. fixed point), and a suitable architectural layout (e.g. bit sliced architecture) determined. Then the lower level design and chip layout must be done.

An alternative to special-purpose chip implementation of the algorithms is the use of the signal processing chips that have recently become available. These are specialized processors that can perform fast floating point operations. Takanashi [106] has implemented the Newton-Euler equations on one of them (the μDP77230) and can calculate the inverse dynamics equations for a six-link, rigid manipulator in 0.9 milliseconds. The forward dynamics equations took 1.9 milliseconds on the processor. It seems that one processor would not be fast enough for implementation of the flexible dynamics algorithms , but it might be feasible to do so using a distributed system employing a few of these processors. This is certainly worth consideration and can be quite interesting.

REFERENCES

1. W. Sunada and S. Dubowsky, "The Application of Finite Element Methods to the Dynamic Analysis of Spatial and Coplanar Linkage Systems", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 103, July, 1981, pp.643-651.
2. W.J. Book and M. Majette, "Controller Design for Flexible Distributed Parameter arms via Combined State-Space and Frequency Domain Techniques", *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*, Vol.105, December, 1983,pp.245-254.
3. R.H. Cannon and E Schmitz, "Initial Experiments on the End-Point Control of a Flexible One-Link Robot", *The International Journal of Robotics Research*, Vol.3, No.3, Fall, 1984, pp.62-75.
4. W.J. Book, "Recursive Lagrangian Dynamics of Flexible Manipulators", The International Journal of Robotics Research, Vol. 3, No.3, Fall, 1984, pp.87-101.
5. T.E. Alberts, G.G. Hastings and W.J. Book, "Experiments in Optimal Control of a Flexible Arm with Passive Damping", *VPI and SU/AAIA Symposium on Dynamics and Control of Large Flexible Structures*, Blackburg, VA, June 1985.
6. P.B. Usoro, R Nadira and S.S. Mahil, "A Finite Element/Lagrange Approach to Modeling Lightweight Flexible Manipulators", *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*", Vol. 108, 1986, pp.198-205.
7. D.A. Streit, C.M Krousgrill and A.K. Bajaj, "A Preliminary Investigation of the Dynamic Stability of Flexible

Manipulators Performing Repetitive Tasks", *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control*, Vol.108, Sept., 1986, pp.206-214.

8. G.G Hastings and W.J. Book, "A Linear Dynamic Model for Flexible Robotic Manipulators", *IEEE Control Systems Magazine*, Vol. 7, No. 1, February, 1987, pp.61-64.
9. G. Naganathan and H.H. Soni, " Coupling effects of Kinematics and Flexibility in Manipulators", *The International Journal of Robotics Research*, Vol. 6, No. 1, Spring, 1987, pp.75-84.
10. J.H. Davis and R.M. Hirschorn, "Tracking Control of a Flexible Robot Link", *IEEE Transactions on Automatic Control*, Vol. 33, No.3, March, 1988, pp.238-248.
11. J.J. Abou-Hanna, "Dynamics and Control of Flexible Manipulators", *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, 1986, pp.470-475.
12. W.J. Book, O. Maizza-Neto and D.E. Whitney, "Feedback Control of Two Beam, Two Joint Systems With Distributed Flexibility", ASME Journal of Dynamic Systems, Measurement and Control, Vol. 97, No. 4, December 1975, pp.424-430.
13. Y. Sakawa, F. Matsuno and S. Fukushima, "Modeling and Feedback Control of a Flexible Arm", *Journal of Robotic Systems*, Vol.2, No.4, 1985, pp.453-472.
14. E. Bayo, " An Improved Numerical Technique for the Control of the Tip Motion of One-Link Flexible Robot", *Proceedings of the Robots 11, 17th International Conference on Industrial Robots*, Chicago, April 26-30, 1987, pp.18-29 to 18-43.

15. I.Y. Shung and M. Vidyasagar, "Control of a Flexible Robot Arm with Bounded Input: Optimum Step Responses", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation,* 1986, pp.916-922.

16. R. Marino and M.W. Spong, "Nonlinear Control Techniques for Flexible Joint Manipilators: A Single Link Approach", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation,* 1986, pp.1030-1036.

17. T. Fukuda, "Flexibility Control of Elastic Robot Arms" *Journal of Robotic Systems,* vol.2, no.1, 1985, pp.73-88.

18. Duffy J.: "Analysis of Mechanisms and Robot Manipulators", (Book),Edward Arnold, 1980.

19. R.P. Paul, "Robot Manipulators: Mathemetics, Programming and Control", The MIT Press, 1981.

20. Denavit J. and Hartenberg R.S.: "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", J. Applied Mechanics, June, 1955.

21. *R.P Paul, B* Shimano and *G* Mayer, "Kinematic Control Equations for Simple Manipulators", IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-11, no. 6, June, 1981.

22. M. Benati, P. Morasso and V. Tagliasco, "The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms", *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control,* Vol.104, 1982, pp.110-113.

23. J.M. Hollerbach and G. Sahar,"Wrist-Partitioned Inverse Kinematic Accelerations and Manipulator Dynamics", *International Journal of Robotics Research,* Vol.2, No.4,

1983, pp.61-76.

24. T. Lozano-Perez, "Task Planning", in *Robot Motion: Planning and Control*, ed J.M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Perez, and M.T. Mason, MIT Press, Cambridge, Mass., pp.463-488.

25. J.M. Hollerbach, "Optimum Kinematic Design for a Seven Degree of Freedom Manipulator", in *Robotics Research: The Second International Symposium*, ed. H Hanafusa and H. Ionoue, MIT Press, Cambridge, Mass., 1985, pp.215-222.

26. C.S.G. Lee and P.R. Chang, "A Maximum Pipelined CORDIC Architecture for Inverse Kinematic Position Computation", *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 5, October, 1987, pp.445-458.

27. T. Watanabe et al., "Improvement of the Computing Time of Robot Manipulators Using a Multiprocessor", *Proceedings of the ASME Winter Annual Meeting*, Miami, Florida, 1985.

28. M. Renaud, "Geometric and Kinematic Models of a Robot Manipulator: Calculation of the Jacobian Matrix and its inverse", *Proceedings of the 11,th International Symposium on Industrial Robots*, Tokyo, Japan, October, 1981.

29. D.E. Orin and W.W. Schrader, "Efficient Computation of the Jacobian for Robot Manipulators", *First International Symposium on Robotics Research*, Breton Woods, New Hampshire, August, 1983.

30. D.E. Orin, K.W. Olson and H. Chao, "Systolic architectures for Computation of the Jacobian for Robot Manipulators", Chapter 3 in *Computer Artchitectures for Robotics and Automation*, ed. J.H. Graham, (Gordon and Breach), 1987.

31. D.E. Whitney, "Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Trans. on Man-Machine Systems, vol. MMS-10, no. 2, June 1969, pp47-53.

32. D.E. Whitney, "The Mathematics of Coordinated Control of Prosthetic arms and Manipulators", *ASME Journal of Dynamic Systems, Measurement and Control*, Vol. 94, Dec. 1972, pp.303-309.

33. J.Y.S. Luh, M.W. Walker and R.P. Paul," Resolved Acceleration Control of Mechanical Manipulators", *IEEE Transactions on Automatic Control*, vol. AC-25, no. 3, June, 1980, pp.468-474.

34. C. Wu and R.P. Paul, "Resolved Motion Force Control of Robot Manipulators", *IEEE Transactions on Systems, Man and Control*, vol. SMC-12, no. 3, May/June, 1982, pp.266-275.

35. C.S.G. Lee, M.J. Chung and B.H. Lee, "Adaptive Control for Robot Manipulators in Joint and Cartesian Coordinates", *Proceedings of the 1983 IEEE International Conference on Robotics and Automation*, 1983.

36. L. Meirovitch, "Elements of Vibration Analysis", McGraw-Hill, 1975.

37. A. Midha, A.G. Erdman and D.A. Frorib, "Finite-Element Approach to Mathematical Modeling of High-Speed Elastic Linkages", *Mechanism and Machine Theory*, Vol. 13, 1978, pp.603-618.

38. D.C. Zienkiewicz, "The Finite Element Method in Engineering Science", McGraw-Hill, London, 1971.

39. R.P. Paul, "Robot Manipulators: Mathematics, Programming and Control",(Book), MIT Press, 1981.

40. A.C. Bejczy, "Robot Arm Dynamics and Control", JPL Technical Memo.33-669, Feb. 1974.

41. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-10, NO. 11, November 1980, pp.730-736.

42. D.E. Orin, R.B. McGhee, M. Vukobratovic and G. Hartoch, "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods", *Mathematical Biosciences*, Vol. 43, No. 1/2, February, 1979, pp.107-130.

43. J.Y.S. Luh, M.W. Walker and R.P. Paul, "On-Line Computational Scheme for Mechanical Manipulators", *Transactions of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 102, June, 1980, pp.67-76.

44. M.W. Walker and D.E. Orin, "Efficient Dynamic Conputer Simulation of Robotic Mechanisms", ASME Journal of Dynamic Systems, Measurement and Control, September 1982, pp.205-211.

45. T.R. Kane and C.F. Wang, "On the Derivation of Equations of Motion", *Journal of the Society of Industrial and Applied Mathematics*, vol.13, 1965, pp.487-492.

46. D.F. Rosenthal and M.A. Sherman, "Symbolic Multibody Equations via Kane's Method", *AAS/AAIA Astrodynamics Specialist Conference*, Paper 83-808, Lake Placid, New York, 1983.

47. H.B. Olsen and G.A. Bekey, "Identification of Robot Dynamics", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986, pp.1004-1010.

48. T. Kubo, G. Anwar and M. Tomizuka, "Application of Nonlinear Friction Compensation to Robot Arm Control", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation,* 1986, pp.722-727.

49. B. Borovac, M. Vukobratovic and D. Stokic, "Analysis of the Influence of Actuator Model Complexity on Manipulator Control Synthesis", *Mechanism and Machine Theory*, Vol. 18, 1981, pp.113-122.

50. M.H. Raibert and B.K.P. Horn: "Manipulator Control using the Configuration-Space Method", The Industrial Robot, vol. 5, no. 2, June, 1978, pp69-73.

51. L Vecchio, S. Nicosia, F. Ficolo and D. Lentini, "Automatic Generation of Dynamical Models of Manipulators", *Proceedings of the Tenth International Symposium on Industrial Robots*, Milan, Italy, March 5-7, 1980, pp.293-301.

52. R. Toogood, "Symbolic Generation of Robot Dynamics Part 1: The DYNAM/CLEAR System", Alberta Centre for Machine Intelligence and Robotics Technical Report No. 87-04, University of Alberta, Alberta, Canada, 1987.

53. J.Y.S. Luh and C.S. Lin, "Automatic Generation of Dynamical Equations for Machanical Manipulators", *Proceedings of the 1981 Joint Automatic Control Conference*, Charlottesville, Va., June 17-19, 1981, pp.TA-2D/1-5.

54. C.P. Neuman and J.J. Murray, "Computational Robot Dynamics: Foundations and Applications", *Journal of Robotic Systems*, Vol. 2, No. 4, 1985, pp.425-452.

55. J.L. Turney and T.N. Mudge, "VLSI implementation of a Numeric

Processor for Robotics", *Proc. of the 27-th International Instrumentation Symposium*, Indianapolis, Indiana, April, 1981.

56. J.Y.S. Luh and C.S. Lin, "Scheduling of Parallel Computation for a Computer-Controlled Mechanical Manipulator", *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-12, No.2, March/April, 1982.

57. C.S.G. Lee, T.N. Mudge and J.L. Turney,"Hierarchical Control Structure Using Special-Purpose Processors for Control of Robot Arms", *Proc. IEEE conf. on Pattern Recognition and Image Processing*, Las Vegas, Nevada, June 14-17, 1982.

58. H. Kasahara and S. Narita, "Parallel Processing of Robot-Arm Control Computation on a Multimicroprocessor System", *IEEE Journal of Robotics and Automation*, Vol. RA-1, No. 2, June, 1985, pp.104-113.

59. R. Nigam and C.S.G. Lee, "A Multiprocessor-Based Controller for the Control of Mechanical Manipulators", *IEEE Journal of Robotics and Automation*, Vol. 1, No. 4, December, 1985, pp.173-182.

60. E.E. Binder and J.H. Herzog, "Distributed Computer Architecture and Fast Parallel Algorithms in Real-Time Robot Control", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-16, No. 4, July/August, 1986, pp.543-549.

61. R.H. Lathrop, "Parallelism in Manipulator Dynamics" *The International Journal of Robotics Research*, Vol. 4, No. 2, pp.80-102, Summer 1985.

62. C.S.G. Lee, and P.R. Chang, "Efficient ParallelAlgorithm for

Robot Inverse Dynamics Conputation", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-16, No. 4, pp.532-542, July/August, 1986.

63. M.W. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", *Proceedings of the 1981 Joint Automatic Control Conference*, The American Automatic Control Council, June, San Fransisco, Ca. 1983.
64. K.R. Symon, "Mechanics", Addison-Wesley, 1971.
65. W.W. Armstrong. "Recursive Solution to the Equations of Motion for an N-Link Manipulator", *Proceedings of the 5,th World Congress on the Theory of Machines and Mechanisms*, Vol. 2, Montreal, PQ, Canada.
66. R. Featherstone, "The Calculation of Robot Dynamics Using Articulated Body Inertias", *International Journal of Robotics Research*, Vol. 2, No. 1, Spring, 1983, pp.13-30.
67. G. Rodriquez, "Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 6, December, 1987.
68. A.E. Bryson, Jr. and Y.C. Ho, "Applied Optimal Control", Blaisdel, 1969.
69. C.S.G. Lee and P.R. Chang, "Efficient Algorithms for Robot Forward Dynamics Computation", *Proceedings of the IEEE International Conference on Robotics and Automation*, Raleigh, NC, March, 1987.
70. P.C. Hughes, "Dynamics of a Flexible Manipulator Arm for Space Schuttle", Paper 28, *Proceedings of the American Astronautical Society and American Institute of Aeronautics and Astrodynamics Specialist Conference*, Jackson Hole,

Wyoming, Sept. 7-9, 1977.

71 R.P. Singh and P.W. Likins, "Manipulator Interactive Design with Interconnected Flexible Elements" *Proceedings of the 1983 Automatic Control Conference*, The American Control Council, San Fransisco, Ca., June, 1983.

72. R.W. Daniel and P.G. Davey, "Two Key Problems in Robotics Research", Printed in *Robotics Research, The Second International Symposium*, MIT Press, 1985, pp.495-499.

73. J.Y.S. Luh, "Conventional Controller Design for Industrial Robots - A Tutorial", *IEEE Transactions on Systems, Man and Cybernetics*, vol.SMC-13, no.3, May/June, 1983, pp.298-316.

74. R.B. Markiewicz, "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer Controlled Manipulator", JPL Tech. Memo 33-601, Mar. 1973.

75. E. Freund, "Direct Design Methods for the Control of Industrial Robots", *Computers in Mechanical Engineering*, Vol. 1, No.4, 1983, pp.71-79.

76. E. Freund, "Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators", *International Journal of Robotics Research*, Vol. 1, 1982, pp.65-78.

77. T.J. Tarn, A.K. Bejczy, A.I. Isidori and Y. Chan, *Prodeedings of the 23'rd Conference on Decision and Control*, 1984, pp.737-751.

78. D.E. Whitney and C.A. Lozinski, "Industrial Robot Calibration Methods and Results", *Proceedings of the International Computers in Engineering Conference*, ASME, Atlanta, Ga.,

August, 1984.

79. C.P. Neuman and P.K. Khosla, "Identification of Robot Dynamics: An Application of Recursive Estimation", *Proceedings of the Fourth Yale Workshop on Application of Adaptive Systems Theory*, K.S. Narendra, editor, Yale University, New Haven, CT, 1984, pp.42-49.

80. G.L. Luo and G.N. Saridis, "Optimal/PID Formulation for Control of Robotic Manipulators" *IEEE International Conference on Robotics and Automation*, 1985, pp.621-626.

81. V.D. Tourassis and C.P. Neuman, "Robust Nonlinear Feedback Control for Robotic Manipulators", IEE Proceedings, Vol. 132, pt.D, No.4, July, 1985, pp.134-143.

82. S.Dubrowsky and D.T. DesForges, "The Application of Model Referenced Adaptive Control to Robotic Manipulators", Trans. ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 101, September, 1979, pp.193-200.

83. A.J. Koivo and T. Guo, "Adaptive Linear Controller for Robotic Manipulators", *IEEE Transactions on Automatic Control*, Feb. 1983.

84. M.E. Kahn and B. Roth, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", *Trans. ASME, Journal of Dynamic Systems, Measurement and Control*, 1971.

85. A. Ficolo, R. Marino and S. Nicosia, "A Composite Control Strategy for a Weakly Elastic Robot", *Proceedings of the 22'nd Allerton Conference*, Urbana, Ill., 1984, pp.225-232.

86. T.E. Alberts, S.L. Dickerson and W.J. Book, "Modeling and Control of Flexible Manipulators", *Robots 9 Conf.*, 1985.

87. M.J. Balas, "Feedback Control of Flexible Systems", *IEEE*

Transactions on Automatic Control, Vol. AC-23, no. 4, August 1978, pp.673-679.

88. A. Truckenbrodt, "Truncation Problems in the Dynamics and Control of Flexible Mechanical Systems", *Proceedings of the IFAC 8'th World Congress*, Vol. XIX, Kyoto, Japan, August 1984, pp.XIX-60 to XIX-65.

89. W.L. Nelson and D. Mitra, "Load Estimation and Load-Adaptive Optimal for a Flexible Robot Arm", *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, 1986, pp.206-211.

90. D. Meldrum and M. Balas, "Direct Model Reference Adaptive Control Using a Reaction Wheel", Printed in *Recent Trends in Robotics: Modeling, Control and Education*, North Holland, 1986, pp.213-220.

91. S. Cetinkunt, B. Siciliano and W.J. Book, "Symbolic Modeling and Dynamic Analysis of Flexible Manipulators", *Proceedings of the 1986 IEEE International Conference on Systems, Man and Cybernetics*, pp.798-803, October, 1986.

92. D.C. Nemir, A.J. Koivo and R.L. Kashyap, "Pseudolinks and the Self-Tuning Control of a Nonrigid Link Mechanism", *IEEE Transactions on Systems, Man and Cybernetics*, vol.18, no.1, Jan./Feb., 1988.

93. K, Khorasani and M.W. Spong, "Invariant Manifolds and their Application to Robot Manipulators with Flexible Joints", IEEE International Conf. on Robotics and Automation, St. Louis, Missouri, 1985.

94. M.W. Spong, K. Khorasani and P Kokotovic, "A Slow Manifold Approach to Feedback Control of Nonlinear Flexible

Systems", *Proceedings of the 1985 American Control Conference*", pp. 1386-1391, Boston, Ma., 1985.

95. N.G. Chalhoub and A.G. Ulsoy, "Dynamic Simulation of a Flexible Robot Arm and Controller", *ASME Journal of Dynamic Systems, Measurement and Control*, June, 1986.

96. N.G. Chalhoub and A.G. Ulsoy, "Control of a Flexible Robot Arm: Experimental and Theoretical Results", *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 109, December, 1987, pp. 299-309.

97. H. Kwakernack and R. Sivan, "Linear Optimal Control Systems", J. Wiley, New York, 1972.

98. J. King, V.G. Gourishankar and R.E. Rink, "Lagrangian Dynamics of Flexible Manipulators Using Angular Velocities Instead of Transformation Matrices", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-17, NO. 11, November/December, 1986, pp. 1059-1068.

99. A Cugy and K Page, "Industrial Robot Specifications", Hermes Publishing, 1984.

100. H.S. Stone, "Introduction to Computer Architecture", Science Reaserch Associates, 1975.

101. K. Hwang and F.A. Briggs, "Computer Architecture and Parallel Processing", New York: McGraw-Hill, 1984.

102. H.T. Kung, "Why Systolic Architecture?", *IEEE Computer*, January, 1982, pp. 37-46.

103. C.A. Mead and L.A. Conway, "Introduction to VLSI Systems", Addison-Wesley, Reading, Massachusetts, 1980.

104. H.T. Kung, "Special-Purpose Devices for Signal and Image Processing: An Opportunity in VLSI", Proceedings of the

SPIE, vol. 241, Real-Time Signal Processing III, July, 1980, pp. 76-84.

105. Y. Huang and C.S.G. Lee, "Generalization of Newton-Euler Formulation of Dynamic Equations to Nonrigid Manipulators", Proceedings of the American Control Conference, Minneapolis, June 10-12, 1987, pp. 72-77.

106. N. Takanashi, "A General Purpose Numerical Calculation Engine for Dynamic Motion Control and Simulation in Manipulators", *Proceedings of the Robots 11, 17th, International Conference on Industrial Robots,* Chicago, April 26-30, 1987, pp. 5-45 to 5-56.